

UNIVERSIDAD POLITÉCNICA DE MADRID

ETSI Informáticos

TRABAJO FIN DE CARRERA

Recubrimiento software e  
interfaz web  
para BADELE+

AUTOR: Javier Polo Araujo

TUTORA: Guadalupe Aguado de Cea

COTUTOR: José Ángel Ramos Gargantilla

FECHA: Octubre 2014



# ÍNDICE

<b>ÍNDICE.....</b>	<b>1</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>3</b>
<b>ACRÓNIMOS Y ABREVIATURAS.....</b>	<b>5</b>
<b>1. INTRODUCCIÓN.....</b>	<b>7</b>
1.1 Alcance del documento.....	8
1.2 Estructura del documento.....	8
<b>2. ASPECTOS BÁSICOS DE LA TEORÍA SENTIDO TEXTO.....</b>	<b>11</b>
2.1. Teoría Sentido –Texto.....	12
2.1.1. Léxico y unidades léxicas.....	13
2.1.2. Funciones léxicas.....	13
<b>3. HERRAMIENTAS EXISTENTES QUE USAN LA TST.....</b>	<b>17</b>
3.1. Herramientas existentes públicamente accesibles.....	18
3.2. BADELE.....	29
3.3. Puntos débiles de BADELE+.....	30
<b>4. PLANTEAMIENTO.....</b>	<b>31</b>
4.1. Objetivos.....	32
4.2. Alcance.....	32
4.3. Objetivos tecnológicos.....	33
4.4. Visión general de la solución.....	33
<b>5. ESPECIFICACION DE REQUISITOS.....</b>	<b>35</b>
5.1. Introducción.....	36
5.1.1. Propósito.....	36
5.1.2. Ámbito del sistema.....	36
5.1.3. Definiciones, acrónimos y abreviaturas.....	36
5.1.4. Visión general del documento.....	36
5.2. Descripción general.....	37
5.2.2. Características de los usuarios.....	37
5.3. Requisitos.....	38
5.3.1. Requisitos funcionales.....	38
5.3.2. Requisitos de arquitectura.....	41
5.3.3. Requisitos de interfaz.....	41
<b>6. DISEÑO Y DESARROLLO.....</b>	<b>43</b>
6.1. Casos de uso.....	44
6.2. Diagrama de clases.....	47
6.3. Diagramas de secuencia.....	48
6.3.1. ObtenerColocaciones.....	48
6.3.2. ObtenerRaiz.....	49
6.3.3. InsertarDefinicion.....	50
6.4. Implementación: organización y tecnologías.....	52
6.5. Desarrollo de la capa de acceso a base de datos.....	54
6.6. Diagrama de clases de la implementación.....	56
6.7. Conversión de datos.....	58
6.8. Interfaz.....	60
6.8.1. Vista.....	60
6.8.2. Accesibilidad de la interfaz.....	62
6.8.3. Modelo.....	63
<b>7. RESULTADOS.....</b>	<b>65</b>
7.1 Resultados.....	66

7.2 Experimentación.....	66
7.3 Características adicionales de usabilidad de la web.....	70
7.4 Comparativa con otras herramientas.....	72
<b>8. LINEAS FUTURAS.....</b>	<b>75</b>
8.1. Perfilado de usuarios.....	76
8.2. Interfaz para dispositivos móviles.....	76
8.3. Mejora de la entrada de información.....	77
<b>9. BIBLIOGRAFÍA.....</b>	<b>79</b>
<b>ANEXO I – OPERACIONES DISPONIBLES EN EL SERVICIO.....</b>	<b>81</b>
<b>ANEXO II – INSTALACIÓN DEL SISTEMA.....</b>	<b>85</b>
10.1. Instalación de los productos.....	86
10.2. Carga de base de datos.....	87
10.3. Despliegue de las aplicaciones.....	89
10.3.1. Modificar la configuración de la aplicación web.....	90
10.3.2. Modificar la configuración del servicio.....	91

# ÍNDICE DE FIGURAS

Figura 3.1 -Interfaz de DiCoInfo.....	18
Figura 3.2 -Interfaz avanzada de DiCoInfo.....	19
Figura 3.3 -Resultados de la búsqueda.....	20
Figura 3.4 -Pantalla inicial de DiCouèbe.....	21
Figura 3.5 -Interfaz de búsqueda de DiCouèbe.....	22
Figura 3.5 -Interfaz de búsqueda de DiCouèbe.....	22
Figura 3.6 -Interfaz de búsqueda de DiCouèbe.....	23
Figura 3.7 -Pantalla inicial de DiCE.....	24
Figura 3.8 -Acceso al diccionario de DiCE.....	25
Figura 3.9 -Resultado de una búsqueda en DiCE.....	25
Figura 3.10 -Busqueda avanzada en DiCE.....	26
Figura 4.1 -Vista general de la solución.....	34
Figura 6.1 -Diagrama de uso del servicio.....	44
Figura 6.2 -Diagrama de casos de uso iniciados por el usuario.....	45
Figura 6.3 -Diagrama de casos de uso iniciados por una aplicación externa.....	46
Figura 6.4 -Diagrama de clases.....	47
Figura 6.5 -Diagrama de secuencia ObtenerColocaciones.....	48
Figura 6.6 -Diagrama de secuencia ObtenerRaiz.....	49
Figura 6.7 -Diagrama de secuencia InsertarDefinicion.....	50
Figura 6.8 – Arquitectura del servicio.....	52
Figura 6.9 – Capa de acceso a base de datos.....	52
Figura 6.10 – Diagrama de comunicaciones.....	54
Figura 6.11 – Diagrama de clases del servicio.....	56
Figura 6.12 – Resultado de una consulta en MSAccess.....	58
Figura 6.13 – Resultado de una consulta en MySQL.....	59
Figura 6.14 – Pantalla inicial del interfaz.....	61
Figura 6.15 – Resultados de una consulta.....	62
Figura 6.16 – Interfaz en un dispositivo portatil.....	62
Figura 6.17 – Diagrama de clases del interfaz.....	64
Figura 7.1 – Vista general del sistema.....	66
Figura 7.2 – Resultados de la busqueda de colocaciones.....	67
Figura 7.3 – Resultados de la consulta de información.....	67
Figura 7.4 – Colocaciones de la palabra coche.....	68
Figura 7.5 – Entorno de desarrollo con un programa de pruebas.....	69
Figura 7.7 – Resultados de la consulta de hiperónimos de la palabra 'coche'.....	70
Figura 7.8 – Ayuda de funciones léxicas.....	71
Figura 7.8 – Ayuda de funciones léxicas.....	71
Figura 7.9 – Ayuda de una función en concreto.....	71
Figura 10.1 – Pantalla inicial del cliente de base de datos.....	87
Figura 10.2 – Formulario de nuevo alias.....	88
Figura 10.3. – Cliente de base de datos listo para usar.....	89
Figura 10.4. – WSDL del servicio instalado.....	90



# ACRÓNIMOS Y ABREVIATURAS

A continuación se lista por orden alfabético los acrónimos y abreviaturas utilizados a lo largo del documento, con una breve explicación del su significado.

API: *Application Programming Interface*, especificación de cómo un componente software permite que otros componentes interactúen con él.

ASCII: *American Standard Code for Information Interchange*, esquema de codificación de caracteres basado en el alfabeto Inglés y que incluye los 128 caracteres más utilizados en ese idioma.

HTML: *HyperText Markup Language*, lenguaje utilizado para la creación de contenidos web.

JDBC: *Java DataBase Connectivity*, mecanismo propio de la plataforma Java para conectar con bases de datos.

MVC: *Modelo Vista Controlador*, patrón de arquitectura de software basado en tres capas diferenciadas, cada una de las cuales realiza unas tareas específicas concretas.

SOAP: *Simple Object Access Protocol*, especificación del formato de los mensajes en comunicaciones entre dos sistemas software.

SQL: *Structured Query Language*, lenguaje que permite definir el acceso a datos en esquemas dentro de bases de datos relacionales.

TST: *Teoría Sentido Texto*.

URL: *Unified Resource Locator*, identificador único de un recurso (que puede ser una página web, una imagen, un fichero de audio u otro elemento) en una red basada en los protocolos que existen en internet.

WEB: *World Wide Web*, conjunto de documentos enlazados unos con otros, accesibles mediante herramientas de navegación web (conocidos como exploradores).

WSDL: *Web Services Description Language*, lenguaje que permite especificar las funcionalidades que un servicio ofrece a los clientes que tengan acceso.

XML: *eXtensible Markup Language*, conjunto de reglas que permite crear lenguajes cuyos documentos pueden ser leídos tanto por personas como por máquinas.





# **1. INTRODUCCIÓN**

## 1.1 Alcance del documento

El presente documento es la memoria del proyecto **BADELEWeb**, un software desarrollado con el objetivo de mejorar la gestión y el aprovechamiento de la base de datos de contenidos lingüísticos denominada BADELE. Este software ofrece desde el manejo básico de la base de datos a la explotación más detallada mediante la ejecución de consultas sobre funciones lingüísticas. Además, como parte del proyecto se incluye una interfaz que permite el aprovechamiento, por parte de un usuario, de ese recubrimiento.

En este documento se presentan los aspectos teóricos necesarios para el desarrollo del proyecto, y la descripción de la funcionalidad que el software debe tener, así como una descripción técnica del desarrollo realizado. Posteriormente se presentan los resultados obtenidos en el proyecto para terminar con una visión de posibles mejoras que se pueden realizar en un futuro. Como anexo se incluye un manual con instrucciones para la instalación del software por parte de un administrador de sistemas para su utilización posterior.

## 1.2 Estructura del documento

Los capítulos en los que se ha dividido este proyecto son:

**Capítulo 2. Aspectos básicos de la Teoría Sentido Texto.** Presenta los conceptos más básicos de la Teoría Sentido Texto (a partir de ahora TST), que son necesarios para poder realizar el resto del proyecto. Cabe señalar que no se profundiza en la teoría en sí, dado que para el desarrollo del proyecto no es necesario. Además, se explica qué es BADELE 3000 y cuales son las carencias observadas que motivan el desarrollo de este proyecto.

**Capítulo 3. Herramientas existentes que utilizan la TST.** Este capítulo describe las herramientas actualmente disponibles para la explotación de la Teoría Sentido Texto aplicada a distintas lenguas, describiendo las fortalezas y debilidades de cada una de ellas. Se propondrán ejemplos para cada una de las herramientas analizadas, mostrando su funcionamiento, para después resumir las características de todas ellas en una tabla comparativa.

**Capítulo 4. Planteamiento.** Tras analizar las herramientas existentes y sus posibles mejoras, se propone la realización de dos componentes software que ayudarán a la explotación del modelo de datos BADELE actualmente disponible, supliendo las carencias actuales detectadas. Se propone el alcance del desarrollo de estos elementos y se plantea la forma general de llevarlo a cabo.

**Capítulo 5. Especificación de requisitos.** En este capítulo se muestran los requisitos que deberá alcanzar el software desarrollado.

**Capítulo 6. Desarrollo.** Este capítulo expone el proceso de desarrollo del proyecto, que comienza con la especificación de requisitos y continúa con los distintos artefactos obtenidos durante el proceso de desarrollo e implementación del proyecto.

**Capítulo 7. Resultados.** Expuestos ya los fundamentos, objetivos y diseño de la solución que constituye este proyecto, se extraen algunas conclusiones sobre los beneficios logrados

al desarrollar el mismo, además de detallar de qué manera se cubren los objetivos propuestos con los componentes desarrollados.

**Capítulo 8. Líneas futuras.** En este capítulo se describen algunos trabajos y estudios que se pueden realizar posteriormente y que toman como base los resultados obtenidos en este proyecto.

**Capítulo 9. Bibliografía del proyecto.**

**Apéndice I: API del sistema.** Se ha añadido un anexo a este documento en el que se exponen de manera detallada todas las funciones que forman la API del sistema BADELEService, e incluye la explicación de sus parámetros, valores devueltos y las excepciones.

**Apéndice II: Manual de instalación del software desarrollado.**



## **2. ASPECTOS BÁSICOS DE LA TEORÍA SENTIDO TEXTO**

## 2.1. Teoría Sentido –Texto

En este capítulo se hace una breve exposición de la Teoría Sentido-Texto, desarrollando únicamente los aspectos fundamentales de la misma que son necesarios para el desarrollo del software requerido.

En los años 60 existía un gran interés en ambos bloques implicados en la guerra fría en la Traducción Automática, que consiste en el uso de software para traducir de un lenguaje natural a otro<sup>1</sup>. En el bloque ruso, un grupo de investigadores rusos que estaban trabajando en traducción automática encabezados por Igor Mel'cuk (Wikipedia 2010) comenzó a dar forma a una serie de herramientas y a una teoría en la que se realizaba la importancia de la semántica de los elementos que la conformaban, en contra de otras corrientes de la época en las que la semántica quedaba relegada a un segundo plano. Finalmente, en 1965 Aleksandr Žolkovskij e Igor Mel'cuk presentaron en Moscú una primera versión de la Teoría Sentido Texto que creían permitiría optimizar la traducción automática de textos en inglés al ruso (que en aquella época tenía que ser realizado a mano por traductores entrenados).

Posteriormente, esta teoría ha cobrado relevancia al ser aplicada en numerosos trabajos de traducción automática, web semántica, extracción automática de información de textos o aprendizaje a través de medios remotos. Al aplicarse la Teoría Sentido Texto (TST a partir de ahora) a toda esta variedad de tareas, se observó la necesidad de enriquecer los recursos lingüísticos disponibles con información concerniente a cada dominio particular en el que se aplicaba esta teoría, buscándose la manera de realizar este enriquecimiento de manera consistente entre todos los dominios de conocimiento.

La TST opera a partir del principio de que un lenguaje funciona como un mapeo entre un significado o sentido y una representación escrita o fonética de ese sentido (Giraldo, 2012). En cualquier acto de comunicación lingüística se dan tres elementos:

- Un contenido de la comunicación, llamado **sentido**.
- Determinadas formas utilizadas para comunicar ese contenido, llamadas **texto**.
- Una serie de correspondencias entre el sentido y el texto, llamadas **lengua**.

Con estos tres elementos, la TST se define como ‘una teoría lingüística que apunta a la descripción de la correspondencia entre sentido y texto dentro de una determinada lengua por medio de la construcción de modelos formales’. Esta correspondencia es multívoca, esto es, un sentido se puede expresar a partir de varios textos, y un texto puede expresar distintos sentidos.

Esta teoría tiene una serie de características importantes:

- Es universal, fundamentándose en principios generales que se dan en todas las lenguas.
- Es formal, lo que significa que permite construir modelos que pueden ser procesados por sistemas lógicos como los programas informáticos.
- Está orientada hacia la síntesis (esto es, producción de sentidos a textos).

---

<sup>1</sup> Consultado el 15 de Agosto de 2012, de [http://es.wikipedia.org/wiki/Unidad\\_l%C3%A9xica](http://es.wikipedia.org/wiki/Unidad_l%C3%A9xica)

Esta teoría tiene una serie de postulados, de los cuales los dos fundamentales son:

### **Postulado 1.**

Una lengua L es un conjunto finito de correspondencias multívocas entre un conjunto numerable de sentidos y un conjunto numerable de textos.

### **Postulado 2.**

Una lengua L puede ser descrita por medio de un modelo funcional. Un modelo funcional de L es un sistema de reglas formales que simulan el comportamiento lingüístico de los hablantes de L; a este modelo se le denomina Modelo Sentido Texto (MST).

#### **2.1.1. Léxico y unidades léxicas**

La definición de léxico, según el diccionario de la RAE, es:

1 m Diccionario de una lengua

2 m Vocabulario, conjunto de las palabras de un idioma, o de las que pertenecen al uso de una región, a una actividad determinada, a un campo semántico dado, etc.

Tal y como se indica en (ConceptoDefinición/Léxico, 2012), el léxico se podría definir con el inventario de las unidades léxicas que conforman una lengua. Sin embargo, para definir de manera formal qué es una unidad léxica, se especifica a su vez cómo diferenciar unas unidades de otras dentro de un determinado léxico. Se pueden plantear dos enfoques, que se muestran a continuación (Wikipedia/Unidad léxica, 2013).

En el primer enfoque, las unidades léxicas se diferencian entre sí por ser unidades aislables, con significación propia y aspectos fonológicos y ortográficos diferenciados. Este enfoque presenta varios problemas; por ejemplo, hay palabras que no tienen significado autónomo (como las preposiciones), los sonidos en muchos casos se superponen y no hay un grafema universal que separe las palabras.

El otro enfoque simplemente sugiere definir una unidad léxica como cualquier elemento situado entre dos espacios de la cadena escrita, comprenda éste o no otros signos de puntuación. Este enfoque, a pesar de ser extremadamente simplista, es uno de los más aceptados actualmente.

#### **2.1.2. Funciones léxicas**

La TST trabaja con Funciones Léxicas (desde ahora FFLL). Las FFLL (Giraldo, 2012) son elementos formales que sirven para describir la relación entre los miembros de una colocación (el

concepto de colocación se define en el siguiente apartado). Por ejemplo, la función **Oper1** vincula la base, un nombre, con el verbo que se utiliza para expresar que alguien lleva a cabo la acción implicada por la base. La función **Oper1** con la base *amor* cobra los valores *sentir* (amor por alguien), *tener* (amor a alguien) o *profesar* (amor por alguien).

Las FFL son funciones en el sentido matemático, de manera que asocian a una unidad léxica (llamada argumento) otra palabra o grupo de palabras llamadas valores. Por ejemplo tenemos la siguiente relación:

$$\mathbf{Culm}_1(\textit{cancer}) = \textit{invasión}$$

De manera que al argumento *cáncer* la función léxica **Culm1** asocia el valor *invasión*.

Una característica importante de las FFL es que describen estas relaciones léxicas en términos universales, esto es, que son aplicables a cualquier lengua. Esto las convierte en una importante herramienta a la hora de realizar traducciones, así como en el campo de la enseñanza de lenguas extranjeras. Por ejemplo,

$$\mathbf{Oper1}(\textit{atención}) = \textit{prestar} \sim.$$

$$\mathbf{Oper1}(\textit{attention}) = \textit{pay} \sim$$

Por lo tanto, vemos que existe la misma relación entre *atención* y *prestar*, y *attention* y *pay*; esta relación es formalizada por la FFL **Oper1**. Entonces, si quisiéramos realizar la traducción al castellano de *to pay attention*, lo haríamos en dos pasos:

- 1) Traducción de la base de la colocación del inglés al español

$$\textit{attention} ==> \textit{atención}$$

- 2) Utilización de la FFL:

$$\mathbf{Oper}_1(\textit{atención}) = \textit{prestar} \sim, \textit{poner} \sim, \textit{otorgar} \sim, \textit{etc.}$$

Con lo que se tendría la posibilidad de traducir la expresión por varias opciones.



### 2.1.3. Colocaciones

Las colocaciones (Giraldo 2012) son un subconjunto de las unidades fraseológicas que responden a pautas de formación gramaticales y su significado es composicional, esto es, se deduce de los significados de los elementos combinados.

Se define unidad fraseológica (Giraldo, 2012) como la combinación de palabras caracterizada por presentar un cierto grado de fijación, de manera que el significado no se deduce necesariamente del significado de los componentes por separado.

A continuación se muestra un ejemplo de lo anterior:

#### **Población activa** (sustantivo + adjetivo)

tiene un significado claramente deducible del significado de cada uno de los elementos que se combinan y que es comúnmente utilizada con un significado ligeramente diferente de la simple unión del significado de los dos elementos; en este caso, de población se obtiene el significado de un grupo de personas amplio y de activa se obtiene el significado de ocupadas en alguna tarea; la unión da un sentido ligeramente distinto al indicar un grupo de personas restringido en el espacio organizativo (añadido este último al significado de la primera palabra) que tiene alguna ocupación laboral.



### **3. HERRAMIENTAS EXISTENTES QUE USAN LA TST**

### 3.1. Herramientas existentes públicamente accesibles

A continuación se mostrarán las herramientas existentes que actualmente hay accesibles por cualquier usuario a través de internet y que permiten consultar recursos organizados siguiendo la TST. De cada uno se describirán las características más reseñables y sus potencialidades. Para cada herramienta se presenta un ejemplo a modo de comparación, si bien no es posible uniformizar esos ejemplos puesto que cada herramienta tiene información muy diferente, además de presentarla también de diferente manera.

#### DiCoInfo<sup>2</sup>

Se trata de un diccionario de términos relacionados con la informática e internet. Se basa en DiCo-OLST, que es una base de datos léxica de francés, desarrollada en el OLST (*Observatoire de Linguistique Sens-Texte*) por Igor Mel'čuk y Alain Polguère, y posteriormente extendida con información en inglés y en castellano (L'HOMME 2009) . Esta aplicación permite realizar búsquedas de un determinado término en tres idiomas diferentes: inglés, francés y castellano, mostrando sinónimos y términos relacionados con la palabra buscada, tal y como se muestra en la figura 3.1.



Figura 3.1 -Interfaz de DiCoInfo

También posee una interfaz más avanzada de búsqueda, aunque se encuentra en una fase de desarrollo muy temprana y no se visualiza correctamente en la mayoría de navegadores, tal y como

<sup>2</sup> <http://olst.ling.umontreal.ca/cgi-bin/dicoinfo/search.cgi>

se muestra en la figura 3.2.

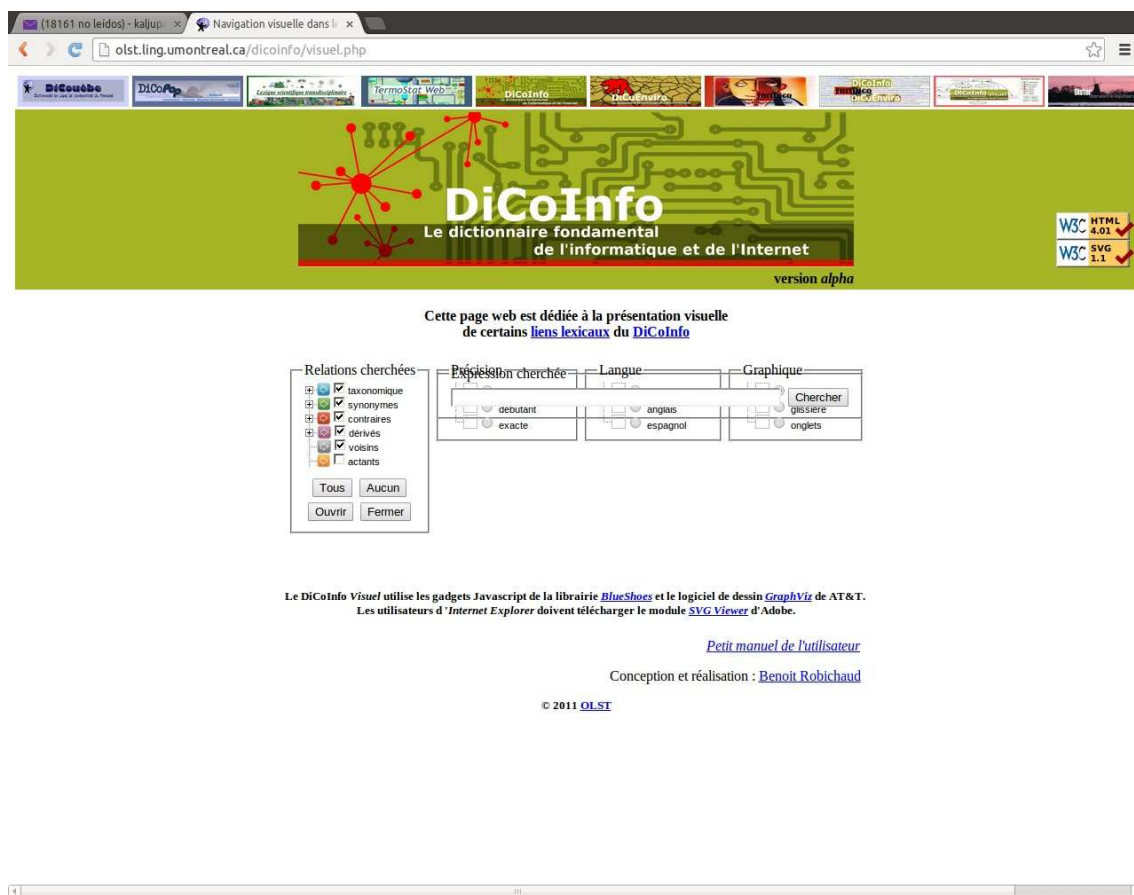


Figura 3.2 -Interfaz avanzada de DiCoInfo

La interfaz esta traducida a tres idiomas: francés, inglés y español. De hecho, la herramienta contiene términos del francés, del inglés y del español, pudiendo realizarse búsquedas en uno de los tres idiomas o en todos a la vez. Estas búsquedas se realizan utilizando cuatro campos para definir las búsquedas:

- Un campo de texto donde escribir el elemento buscado.
- El modo, que permite seleccionar entre palabra, término, relación léxica, función léxica y expresión.
- El idioma del elemento buscado.
- La precisión, esto es, si el elemento buscado es exacto, empieza por el texto introducido en el primer campo o contiene el texto introducido en el primer campo.

Por ejemplo, se realiza una búsqueda con estos parámetros:

- Modo: palabra.

- Idioma: español.
- Precisión: exacto.
- El texto introducido es 'ordenador'

Los resultados obtenidos con esa búsqueda se muestran en la figura 3.3.



Figura 3.3 -Resultados de la búsqueda

Los resultados obtenidos contienen la definición del término buscado, así como otra información y otros términos sinónimos. Además, resulta posible enlazar con parte de los resultados, por ejemplo pulsando sobre 'usuario' se accede a la información de ese término.

Se observa, sin embargo las siguientes carencias. Por un lado, no hay ninguna manera de realizar exportaciones de los resultados a algún formato que sea más fácilmente manejable. Por otro lado, no resulta posible realizar consultas desde otros programas a la información contenida, siendo por tanto el interfaz web el único método de interacción con el sistema. Finalmente, conviene indicar que existe tanto una ayuda en línea como un documento explicativo del manejo de la herramienta, pero ambos se encuentran únicamente en francés.

## Dicouèbe<sup>3</sup>

Se trata de otra aplicación web basada en DoCo-OLST, desarrollada por el *Observatoire de linguistique Sens-Texte*. Esta es un laboratorio de lingüística compuesto por tres entidades: el CNRS, la Escuela Normal Superior y la Universidad Paris 3. Esta herramienta se centra sólo en el idioma francés, y esta completamente escrita en francés. En la figura 3.4 se muestra la pantalla inicial.



Figura 3.4 -Pantalla inicial de DiCouèbe

Posee una interfaz de búsqueda con multitud de opciones, así como la posibilidad de realizar búsquedas combinando varios elementos; desafortunadamente, todo el interfaz esta también en francés, tal y como se muestra en la figura 3.5, por lo que resulta difícil de manejar si no se conoce ese idioma.

<sup>3</sup> <http://olst.ling.umontreal.ca/dicouebe/>

The screenshot shows the DiCouèbe web interface. At the top, there's a header with the logo and the text "LE DICO EN LIQUE". Below the header, a message states "Aucune requête effectuée pour l'instant." (No query performed for the moment). A search bar is present with buttons for "Chercher" (Search), "Chercher nouvelle fenêtre" (Search new window), "Effacer tout" (Clear all), and "Effacer tous les champs" (Clear all fields). There's also a "Mode expert" checkbox.

The main content area is divided into several sections, each with a list of checkboxes and corresponding input fields:

- LEXIE**
  - ☐ nom vocable [lexie:vocable]
  - ☐ no. acception [lexie:num]
  - ☐ est acception de base? [lexie:ub]
  - ☐ carac. grammaticales [lexie:cgs]
  - ☐ est étiquette? [lexie:estEtiquette]
  - ☐ nota bene [lexie:nb]
  - ☐ étiquette sém. [lexie:formuleEtiquette]
  - ☐ struct. actancielle [lexie:formePropositionnelle]
- LIENS DE FONCTIONS LEXICALES**
  - ☐ rubrique de fi [FL:rubriqueFL]
  - ☐ fonction lexicale [FL:formuleFL]
  - ☐ est standard? [FL:estStandard]
  - ☐ glose [FL:glose]
  - ☐ groupe valeur [FL:famille]
  - ☐ gradation [FL:gradation]
  - ☐ valeur fusionnée? [FL:estFusionnee]
  - ☐ marque d'usage [FL:marqueDUsage]
  - ☐ valeur [FL:lexie]
  - ☐ régime valeur [FL:regime]
  - ☐ contrainte [FL:contrainte]
  - ☐ exemple [FL:exemple]
- ACTANTS SÉMANTIQUES**
  - ☐ actant sémantique [asemvar:asemvar]
  - ☐ dépendance [asemvar:dependance]
  - ☐ optionnel [asemvar:optionnel]
  - ☐ étiquette sém. [asemvar:formuleEtiquette]
- ACTANTS SYNTAXIQUES**
  - ☐ num. tableau [asyntp:numTableau]
  - ☐ contraintes tableau [asyntp:contrainteTableau]
  - ☐ actant sémantique [asyntp:asem]
  - ☐ actant syntaxique [asyntp:asynt]
  - ☐ marque d'usage [asyntp:marqueDUsage]

Figura 3.5 -Interfaz de búsqueda de DiCouèbe

Para comparar el funcionamiento de la herramienta, se intenta realizar una búsqueda de palabras en francés. En concreto, se realiza la búsqueda de las siguientes palabras utilizando el campo *vocable*:

ordinateur (ordenador)  
 exemple (ejemplo)  
 voiture (coche)  
 clavier (teclado)  
 femme (mujer)

Ninguna de estas palabras devuelve resultados, aunque resulta interesante notar que se puede visualizar la sentencia SQL ejecutada en cada búsqueda. Para obtener resultados, se realiza una búsqueda diferente, en concreto, de una función léxica (*Caus*). En este caso, sí se obtienen resultados, tal y como se muestra en la figura 3.6.



The screenshot shows the DiCouèbe web interface. At the top, there's a browser address bar with the URL `idexfix.ling.umontreal.ca/dicouebe/main.php?MODE_D_ACCES=acces_standard`. Below the address bar, there's a search form with several sections:

- LEXIE**: Includes checkboxes for `nom vocable`, `no. acception`, `carac. grammaticales`, and `étiquette sém.`.
- LIENS DE FONCTIONS LEXICALES**: Includes checkboxes for `fonction lexicale`, `glose`, `valeur fusion`, `marque d'usage`, `valeur`, `regime valeur`, `contrainte`, and `exemple`.
- LOCUTIONS CONSTRUITES AVEC LE MOT-CL**: Includes checkboxes for `locution` and `exemple`.
- EXEMPLES D'EMPLOI DU MOT-CL**: Includes checkboxes for `phrase d'exemple`.

Below the search form, there's a copyright notice: "Copyright © 2005 OLST—Université de Montréal. Tous droits réservés. <http://idexfix.ling.umontreal.ca/dicouebe/LICENCE.txt>".

The results section shows "772 résultats" and a button "Exporter les résultats". Below this is a table with two columns: "lexie" and "FL". The table contains the following data:

lexie	FL
vocabulaire	formuleFL
BORD	Y tant Individu Caus2Real2
ARRIÈRE-PLAN	CausOper1
ARRIÈRE-PLAN	CausContOper1
ABANDON	CausOper1
ABANDON	CausFunc0
ABATTEMENT(1)	CausOper1
ABÊLLE	Caus1Mult
ABÊLLE	CausFunc0
ABÊLLE	CausReal1
ABÊLLE	CausOper1
ACCUSATION	AntiVer-motif.CausFunc0
ADMIRATION	Caus1Manif
ADMIRATION	Caus1Manif2
ADMIRATION	S0SingCaus1Manif
AFFECTION	CausFunc2
ÂGNEAU	CausFunc0
AMI	X et Y tant plus des enfants, Caus1Oper1
AMI	Caus2Oper2
AMI	Caus2Labor21
AMI	cause d'une action de Y FinOper2
AMPOULE	CausFact0
AMPOULE	Caus1Oper1
ANGOISSE	Caus2PredPlus
ANGOISSE	CausPredMinus
ANGOISSE	CausDe_nouveauFunc0

Figura 3.6 -Interfaz de búsqueda de DiCouèbe

Los resultados obtenidos se obtienen como una lista de elementos exportable a un fichero de texto.

Como principales carencias, se observa que no existen enlaces entre resultados. Tampoco es posible realizar consultas desde otros programas al sistema, con lo que la interfaz web es la única manera de acceder a la información. Y por último, el hecho de que la interfaz (y la documentación de ayuda del sistema) esté únicamente en francés dificulta notablemente su utilización por usuarios no francófonos.

## Dice<sup>4</sup>

Se trata del Diccionadio de Colocaciones del Español, desarrollado por el Grupo DICE de la Universidad de La Coruña. Tal y como se indica en la propia web, se trata de un entorno formado por dos componentes: el propio diccionario y la interfaz de consultas avanzadas (DICE 2012). La pantalla inicial se muestra en la figura 3.7.

4 <http://www.dicesp.com/paginas>



Figura 3.7 -Pantalla inicial de DiCE

En esa pantalla se explica qué es una colocación, a modo de introducción al propio sistema. La interfaz de acceso al diccionario de la figura 3.8 muestra un listado de los lemas disponibles en el sistema para realizar la búsqueda de las unidades léxicas asociadas a cada lema.

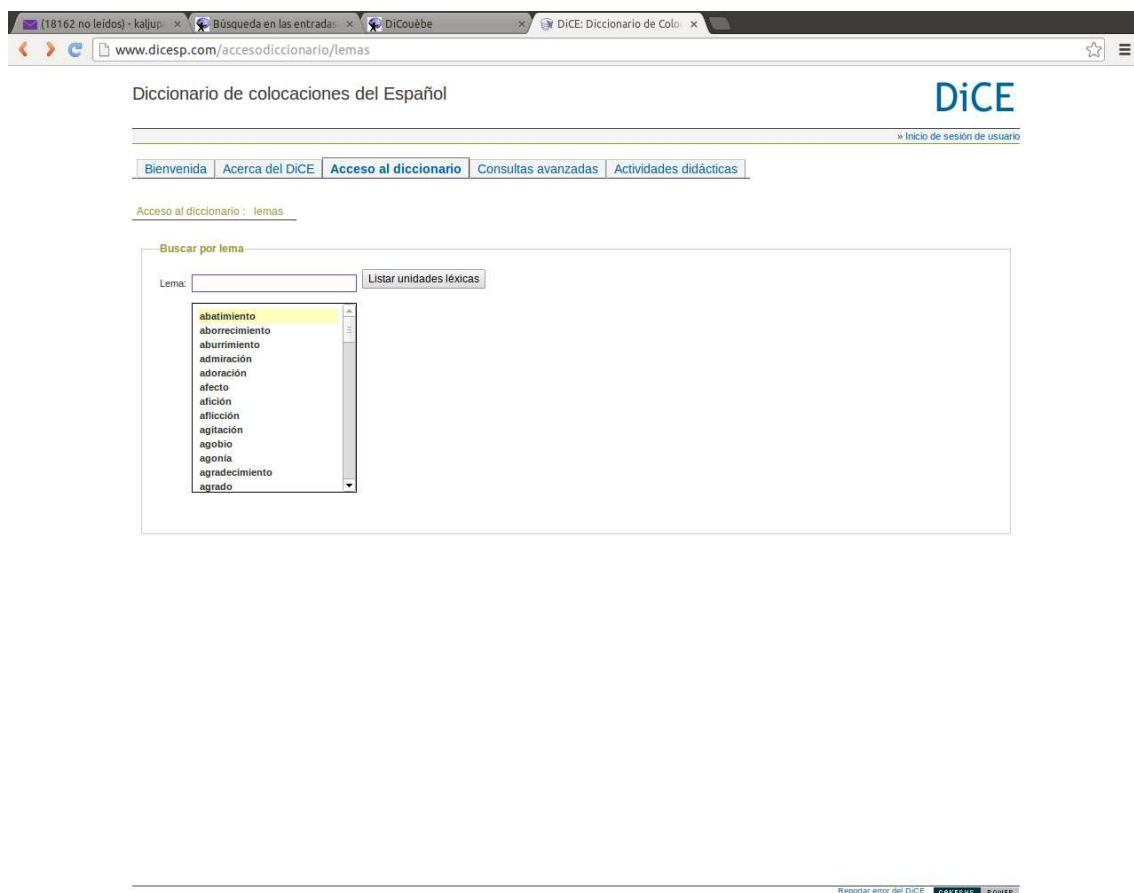


Figura 3.8 -Acceso al diccionario de DiCE

Al realizar una búsqueda, se obtienen las unidades léxicas asociadas al lema seleccionado. A modo de ejemplo, se realiza la búsqueda con el lema **ilusión**, obteniéndose los resultados mostrados en la figura 3.9.

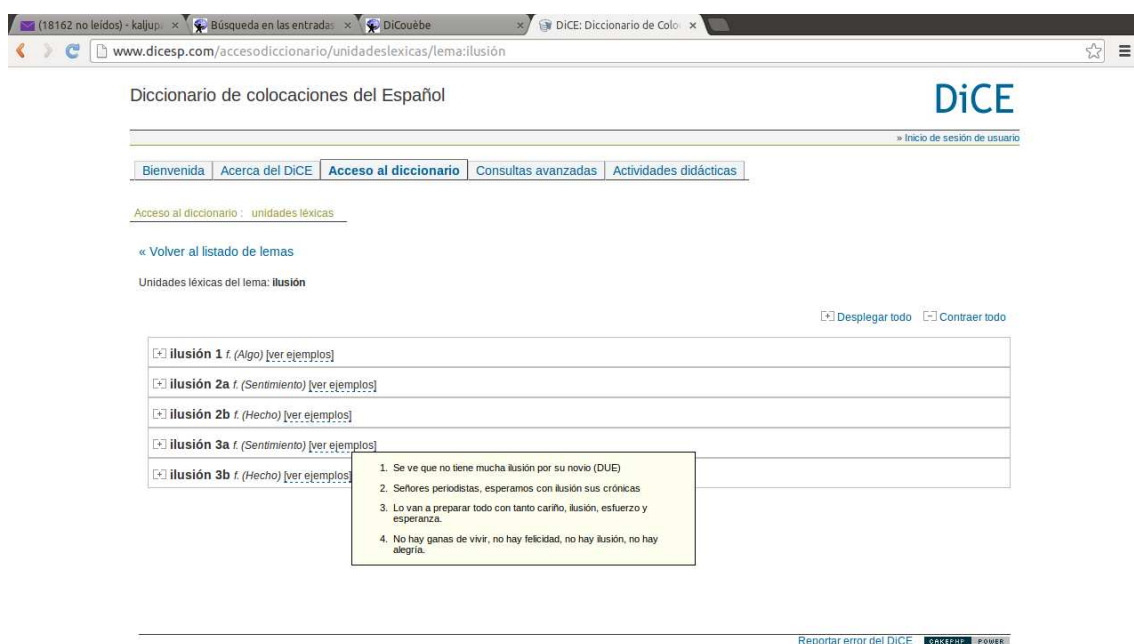


Figura 3.9 -Resultado de una búsqueda en DiCE

De cada uno de los resultados se pueden visualizar ejemplos de uso de los mismos.

Por otro lado, en otras de las pestañas de la aplicación se pueden realizar búsquedas más avanzadas, separadas por los siguientes apartados:

- **¿Qué significa?.** Permite buscar una unidad léxica junto con un colocativo obteniendo una glosa.
- **Ayuda a la redacción.** Permite asociar una unidad léxica con un grupo de palabras de un determinado sentido para obtener de ese grupo una palabra correctamente asociada a la unidad léxica.
- **Búsquedas directas e inversas.** Las dos pestañas permiten obtener o bien valores asociados a una determinada función léxica, o bien funciones léxicas aplicables a un determinado valor.

En la figura 3.10 se puede ver una de las interfaces descritas.



Figura 3.10 -Búsqueda avanzada en DiCE

Una característica interesante del sistema es que ofrece ayuda en línea a lo largo de la interfaz, de manera que cada operación contiene una pequeña indicación del uso de la misma.

La principal limitación observada, similar a los anteriores casos, es la imposibilidad de acceder a la funcionalidad del sistema mediante otros sistemas automáticos, lo cual limita el uso que se puede dar al sistema a consultas en línea de usuarios. Otra limitación es la imposibilidad de realizar exportaciones de los resultados obtenidos.

## Cuadro resumen.

La siguiente tabla muestra a modo de resumen las características de las herramientas estudiadas; los números entre paréntesis indican el número de la observación que se escribe a continuación del cuadro.

Herramienta	DiCoInfo	Dicouèbe	DiCE
<b>Grupo desarrollador</b>	Universidad de Montreal	Laboratorio LaTTICe (3)	Grupo Dice (Universidad de La Coruña)
<b>Dominio del contenido</b>	Informática e internet	Genérico	Genérico
<b>Idiomas contenidos</b>	Francés, Inglés, Español	Francés	Español
<b>Operaciones web</b>	Búsqueda de términos utilizando distintos filtros (1)	Búsqueda de vocablos, funciones léxicas, ejemplos, etc. (3)	Consulta de colocaciones, significado de términos, ayuda a la redacción
<b>Operaciones remotas</b>	No tiene	No tiene	No tiene
<b>Librerías para uso remoto</b>	No tiene	No tiene	No tiene
<b>Enlace entre consultas</b>	Enlaces a términos relacionados	No tiene	Enlaces a colocaciones
<b>Idiomas del interfaz</b>	Francés, Inglés, Español (2)	Francés	Español
<b>Exportación de resultados</b>	No tiene	Permite exportar los resultados a un fichero de texto	No tiene
<b>Ayuda</b>	Contiene un manual de ayuda, así como un resumen, pero únicamente en francés	Contiene un manual únicamente en francés	Posee ayuda en línea en español

Sobre estas características se pueden hacer las siguientes observaciones:

1. Únicamente contempla términos relacionados con la informática e internet.
2. La interfaz esta disponible en esos tres idiomas, sin embargo la ayuda sólo esta disponible en frances.
3. Permite una gran variedad de consultas de elementos de la base de datos, así como combinaciones entre búsquedas (o sea, búsquedas por más de un elemento).  
Posteriormente, en el apartado 7.4 de este documento, se volverá a exponer esta tabla añadiendo una columna con la información relacionada con el software resultado del proyecto, de manera que

se puedan comparar este software desarrollado con las herramientas actualmente existentes.

## 3.2. BADELE

En (Barrios, 2008) se analiza el interés de estudiar el dominio de las FFLL, esto es, el paradigma de las unidades léxicas para las que cada función léxica fue creada, defendiendo la idea de que las propias funciones léxicas imponen restricciones a sus bases (a los elementos sobre los que se aplican); estas restricciones se denominan rasgos de dominio. Para este estudio se hizo necesario recabar un gran número de datos léxicos del español. Toda esta información y la aplicación del principio de herencia léxica al dominio de las funciones léxicas han sido los factores que han orientado el diseño de una base de datos para implementar y posteriormente validar este dominio, denominada BADELE.3000, que contenía en aquel momento los 3300 sustantivos más usados en español y que permitió generar y recoger unas 9000 relaciones léxicas, que fueron posteriormente ampliadas a partir de otros recursos hasta superar las 20000 relaciones formalizadas por medio de funciones léxicas.

Actualmente se dispone de esa base de datos, y de algunas herramientas para la extracción de información de la misma. Sin embargo, se carece de un juego de herramientas que permita esa extracción por parte de otros sistemas remotos (para su uso en trabajos automáticos) así como de una interfaz unificada que sea sencilla y amigable desde el punto de vista de un lingüista que desee explotar la información contenida sin preocuparse del continente de la misma (esto es, interesado en lo que la base de datos le puede dar, pero no en la base de datos en sí).

### 3.3. Puntos débiles de BADELE+

Antes de desarrollar el software que resulta de este proyecto, lo que existe como BADELE es un modelo de datos, con una serie de tablas pobladas por información lingüística y un conjunto de consultas que permiten obtener determinada información de esa base de datos. No tiene, por lo tanto, una capa de software por encima que permita un uso programático o remoto a las consultas existentes, ni una interfaz que permita consultar la información de manera sencilla. Por lo tanto, a modo de enumeración las carencias observadas son las siguientes:

1. El uso de herramientas limitadas a intérpretes de lenguajes basados en SQL, lo que dificulta o limita la interacción con estos sistemas, y la carencia de interfaces de explotación de la información familiares para los lingüistas o para cualquier otro usuario interesado en la información contenida (por ejemplo, un traductor).
2. La inexistencia de herramientas de gestión de la información contenida, esto es, la incapacidad de ampliar o corregir de manera sencilla por parte de los lingüistas la información contenida.
3. La incapacidad de la base de datos de ser explotadas por otros sistemas, sobre todo de manera remota. La explotación de la base de datos por parte de otro software que quiera acceder a ella, debe realizarse accediendo directamente a la base de datos, lo cual resulta sumamente peligroso ya que se expone la propia información que podría ser corrompida accidentalmente o intencionadamente por un software. Además, esta explotación obliga a que la base de datos y el software que quiere utilizarla tengan accesibilidad física entre sí, esto es, deben estar en la misma máquina, o en máquinas separadas en una misma red de datos, por lo que no se podría ofrecer ese servicio, por ejemplo, a otra universidad que quiera acceder a él.



## **4. PLANTEAMIENTO**

En este capítulo se plantean los objetivos que se pretenden alcanzar por el proyecto, así como las limitaciones del mismo. Así mismo, se ofrece una visión general de la solución adoptada.

## 4.1. Objetivos

De acuerdo a las posibilidades de mejora de BADELE+ se plantean los siguientes objetivos:

1. Ocultar la complejidad de la base de datos, tanto de su esquema como de las consultas SQL utilizadas a cualquier usuario del sistema, ofreciendo una interfaz de explotación de la información lo más sencilla e intuitiva posible.
2. Disponer de un conjunto de métodos de gestión de la información que permita corregir y ampliar la información contenida en la base de datos.
3. Facilitar la utilización por parte de otros sistemas de la funcionalidad ofrecida, esto es, que un sistema remoto pueda utilizar las funcionalidades del sistema con las mismas capacidades que un hipotético usuario humano.

## 4.2. Alcance

Para desarrollar los objetivos planteados, se propone construir un sistema con los siguientes componentes y características:

1. Creación de una herramienta para la gestión de la información contenida en la base de datos. Esta herramienta debe ser capaz de ejecutar todas las funciones contenidas en la base de datos, así como la inserción de elementos en la base de datos. Debe ofrecer una API utilizable por otras aplicaciones y herramientas.
2. Diseño de esta herramienta de forma que el sistema sea independiente del gestor de base de datos empleado para su implementación, con el objetivo de que se pueda ampliar el uso de BADELE+ a entornos con otros gestores de base de datos.
3. Creación de una interfaz gráfica para permitir la utilización del sistema de forma interactiva por parte de usuarios no necesariamente lingüistas.
4. Independencia de la plataforma sobre la que se ejecute el sistema, con el mismo objetivo de facilitar el uso del sistema en el mayor número de entornos operativos.
5. Capacidad de acceso multiusuario al sistema, para permitir un uso concurrente del mismo por parte de todos los usuarios interesados.
6. Extensibilidad del sistema. Dado que no es posible trabajar sobre la extensibilidad del modelo de datos, que ya viene diseñado, se debe facilitar la ampliación del sistema para ofrecer nuevas funcionalidades.

### 4.3. Objetivos tecnológicos

Dados los objetivos anteriormente expuestos, y el alcance contemplado, se deben observar los siguientes objetivos tecnológicos:

1. Conversión de BADELE completo de la tecnología actual a otra tecnología de base de datos que permita una más sencilla explotación de la base de datos.
2. Recubrimiento de todas las consultas incluidas actualmente en BADELE mediante métodos programados que permitan su uso por otros componentes software locales o remotos.
3. Ofrecimiento de una interfaz web lo más sencilla posible que permita ejecutar todas las consultas incluidas en BADELE+, así como otras básicas.
4. Ofrecimiento de una interfaz web que permita ejecutar las operaciones de inserción y modificación actualmente existentes en BADELE+.

Las limitaciones del proyecto son las siguientes:

1. Optimización de las consultas; estas consultas son las que actualmente están disponibles sin entrar en la optimización de las mismas.
2. Las funciones léxicas cubiertas son únicamente las implicadas en el listado de consultas indicadas en el anexo I.
3. En general, no se modifica el esquema de base de datos existente ni la información contenida en el mismo, ya que el proyecto trata sobre la explotación, no sobre la mejora del mismo.

### 4.4. Visión general de la solución

Una vez expuestos los objetivos y el alcance del proyecto, se enumeran brevemente las estrategias empleadas para implementar la solución planteada:

1. Creación de una API de alto nivel formada por métodos capaces de explotar la información contenida en la base de datos BADELE+. Esto cubriría el objetivo 1.
2. Creación de un servicio que mediante el recubrimiento de esta API permita la invocación del mismo de forma remota, proporcionando así la posibilidad de que cualquier herramienta que soporte invocación de servicios remotos haga uso de esa API sin necesidad de disponer físicamente de ella (computación remota). Con esto se cubre el objetivo 2.
3. Creación de una interfaz web de usuario para permitir la utilización del sistema de forma interactiva. Con esto se cubre el objetivo número 3.

En la figura 4.1 se ofrece una representación gráfica de la solución:

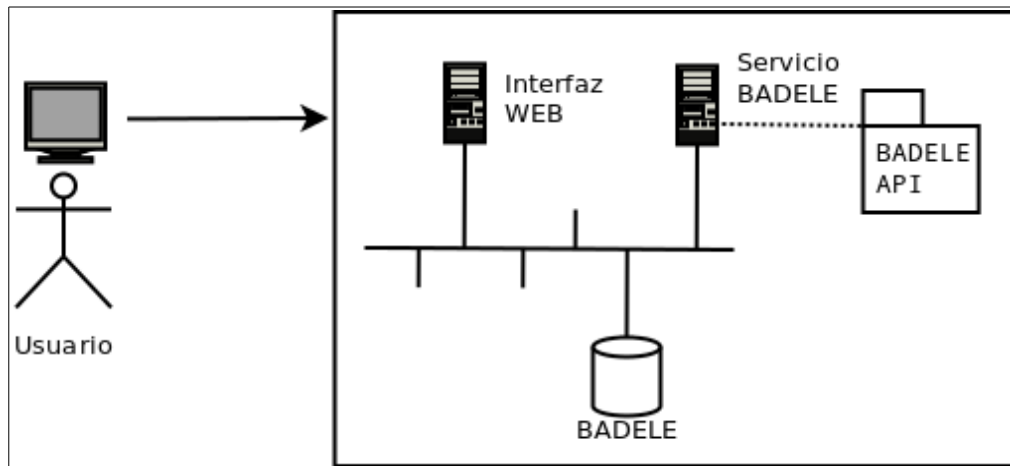


Figura 4.1 -Vista general de la solución

Estos elementos se implementan con las siguientes características software:

1. Desarrollo del software en lenguaje Java; este lenguaje es ampliamente utilizado en la industria y ofrece independencia de la plataforma.
2. En cuanto a la selección del gestor de base de datos, se decide el uso de MySQL por ser el más utilizado, además de disponer de licencias a coste cero para determinados entornos como el académico.
3. Para implementar la extensibilidad del recubrimiento se propone una arquitectura orientada a objetos y basada en capas (separación de acceso a datos, lógica de negocio y presentación), así como la diferenciación de funcionalidades de entrada, salida, gestión y procesamiento de los datos.
4. Como resultado de lo anterior, el software generado se organiza en librerías, que permiten su reutilización en proyectos futuros.

## 5. ESPECIFICACION DE REQUISITOS

## 5.1. Introducción

En este capítulo se describe una Especificación de Requisitos Software (ERS) para el sistema de recubrimiento de BADELE+. Esta especificación se ha estructurado inspirándose en las directrices dadas por el estándar [IEEE Sp 830, 98].

### 5.1.1. Propósito

Este capítulo intenta definir de la manera más precisa y clara posible todas las funcionalidades y restricciones del sistema solicitado. El presente capítulo ha sido confeccionado por el desarrollador y está dirigido tanto a él como a los responsables del OEG (Ontology Engineer Group), constituyéndose como el elemento de comunicación entre ambos y como el acuerdo de definición una vez aprobada su validez por ambas partes. Dicha validez se alcanzará tras las sucesivas revisiones, después de lo cual se convertirá en la base del trabajo posterior del desarrollador.

### 5.1.2. Ámbito del sistema

El OEG desea dotar de accesibilidad, para su explotación, a la información contenida en la base de datos BADELE+, previa migración a un gestor de base de datos más potente que el actual MS Access. De esta necesidad surge la idea de desarrollar un recubrimiento software que ofrezca esa accesibilidad, dotándolo además de una interfaz web que permita un uso más cómodo y eficiente por parte de los usuarios finales.

### 5.1.3. Definiciones, acrónimos y abreviaturas

En este caso no se necesitan definiciones especiales.

### 5.1.4. Visión general del documento

El presente documento, tal y como se puede observar en el índice, está estructurado en tres partes claramente diferenciadas. La primera es esta introducción que describe sus características generales. La segunda describe el sistema de un modo general. Y la tercera parte define detalladamente el conjunto de requisitos que el sistema debe cumplir según petición de los responsables del OEG.

## 5.2. Descripción general

El sistema tendra la capacidad de ofrecer un mecanismo de extracción de información de la base de datos BADELE+ mediante métodos públicos ofrecidos a sistemas remotos, siendo estos metodos obtenidos de la actual base de datos así como otros métodos de uso común. Además, se desarrollará una interfaz web capaz de interactuar con este sistema ofreciendo acceso a estos métodos de uso común, parar facilitar el uso por parte de cualquier persona de los mismos.

### 5.2.1. Funciones del sistema.

El sistema realizará las funciones descritas en los requisitos del Fun #0001 and Fun #0010; por otro lado, las tareas del interfaz se describen en los requisitos Fun #0011 y Fun #0012.

### 5.2.2. Características de los usuarios

El sistema está pensado para que sea manejado tanto por otros programas a través de la interfaz software como por usuarios no informáticos, por lo que la interfaz web ha de ser intuitiva, fácil de aprender y sencilla de manejar.

### 5.2.3. Restricciones

En este caso no aparecen restricciones especiales.

## 5.3. Requisitos

En este apartado se presentan los requisitos que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son esenciales, es decir, sería inaceptable un sistema que no satisfaga alguna de estas funcionalidades.

La especificación de requisitos se divide en:

- Requisitos funcionales.
- Requisitos de arquitectura.
- Requisitos de interfaz

### 5.3.1. Requisitos funcionales

En total, el sistema final debe cumplir 12 requisitos funcionales, que son:

Fun #0001
Prioridad 1
El servicio debe ofrecer un interfaz al exterior que permita ejecutar las funciones incluidas en la base de datos BADELE+. Estas funciones son las disponibles en la versión Access de la base de datos.

Fun #0002
Prioridad 1
Cada función debe tener como parámetros al menos un usuario y una contraseña necesarias para ejecutarla, de manera que sólo los usuarios autorizados tengan acceso a las mismas.

Fun #0003
Prioridad 1
Cada función debe disponer de los parámetros extras para implementar la funcionalidad propia del método.



Fun #0004
Prioridad 1
El servicio debe ofrecer un interfaz al exterior que permita ejecutar las funciones contenidas en BADELE+. Este interfaz debe estar disponible para los sistemas autorizados, que dispondrán de las credenciales descritas en el requisito Fun #0002.

Fun #0005
Prioridad 1
Cada función debe retornar los resultados de la ejecución de una manera consistente entre todas las funciones, esto es, todas las funciones deben utilizar el mismo formato.

Fun #0006
Prioridad 1
El servicio debe ofrecer un interfaz al exterior que permita ejecutar los métodos de actualización de datos contenidos en BADELE+; estos métodos serán obtenidos a partir de la versión Access de BADELE+

Fun #0007
Prioridad 1
El servicio debe interactuar con la base de datos BADELE+ ya definida, utilizando sus tablas y datos, después de que esta base de datos haya sido migrada a un gestor de bases de datos relacionales.

Fun #0008
Prioridad 1
El servicio escribirá los resultados de los métodos de inserción en las tablas de la base de datos BADELE+.

Fun #0009
Prioridad 2
Además de estos métodos y funciones derivados del modelo de BADELE+, se ofrecerá una función para obtener todas las funciones léxicas.

Fun #0010
Prioridad 1
<p>Se implementarán una serie de funciones adicionales, actualmente no disponibles en BADELE+, que permiten realizar consultas comúnmente utilizadas por los lingüistas. Estas funciones son:</p> <ul style="list-style-type: none"> <li>- Definición de una palabra</li> <li>- Información de una palabra</li> <li>- Sinónimos de una palabra</li> <li>- Antónimos de una palabra</li> <li>- Hipónimos de una palabra</li> <li>- Hiperónimos de una palabra</li> <li>- Colocaciones de una palabra</li> <li>- Obtener los valores asociados a una función léxica y una palabra</li> <li>- Obtener los valores asociados a una función léxica</li> </ul>

Fun #0011
Prioridad 1
Se ofrecerá una interfaz web que ofrezca un acceso sencillo a las funciones indicadas en Fun#0010

Fun #0012
Prioridad 1
La interfaz web será capaz de exportar los resultados de una consulta cualquiera en formato MSEXcel

### 5.3.2. Requisitos de arquitectura

Arq #0001
Prioridad 1
El servicio se implementará utilizando un modelo MVC.

Arq #0002
Prioridad 1
El servicio ofrecerá un interfaz accesible de manera remota mediante un cliente SOAP. Este interfaz será la vista del modelo.

Arq #0003
Prioridad 1
Habrà una capa de funcionalidad intermedia asociada a cada método de la vista que será el controlador del modelo MVC.

### 5.3.3. Requisitos de interfaz

El interfaz web viene determinado por los 4 requisitos detallados a continuación:

Int #0001
Prioridad 1
El interfaz será accesible a través de los navegadores Iexplorer (Versión 6 y posteriores), Firefox (17.0.1 y posteriores) y Chrome (23.0 y posteriores).

Int #0002
Prioridad 1
El texto del interfaz estará disponible en dos idiomas, castellano e inglés.

Int #0003
Prioridad 1
Se incluirá en la interfaz la ayuda necesaria para que un usuario no informático pueda hacer uso del sistema con facilidad.

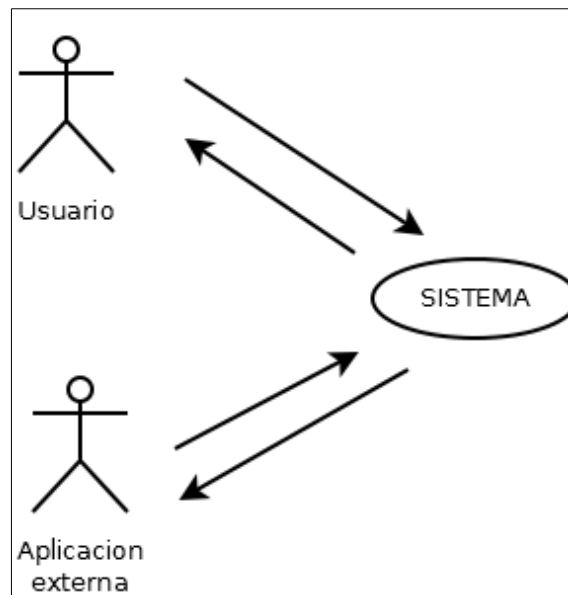
Int #0004
Prioridad 1
La interfaz tendrá un aspecto similar al actualmente utilizado por el resto de páginas del dominio pln.oeg-upm.net.

## 6. DISEÑO Y DESARROLLO

Con el análisis de requisitos presentado en el capítulo 5, se procede al diseño y la implementación de la aplicación, siguiendo los requisitos detallados en el mencionado capítulo, distinguiendo entre el servicio de acceso a la información y la interfaz web.

## 6.1. Casos de uso

En la figura 5.1 se puede ver un diagrama general de uso del servicio.

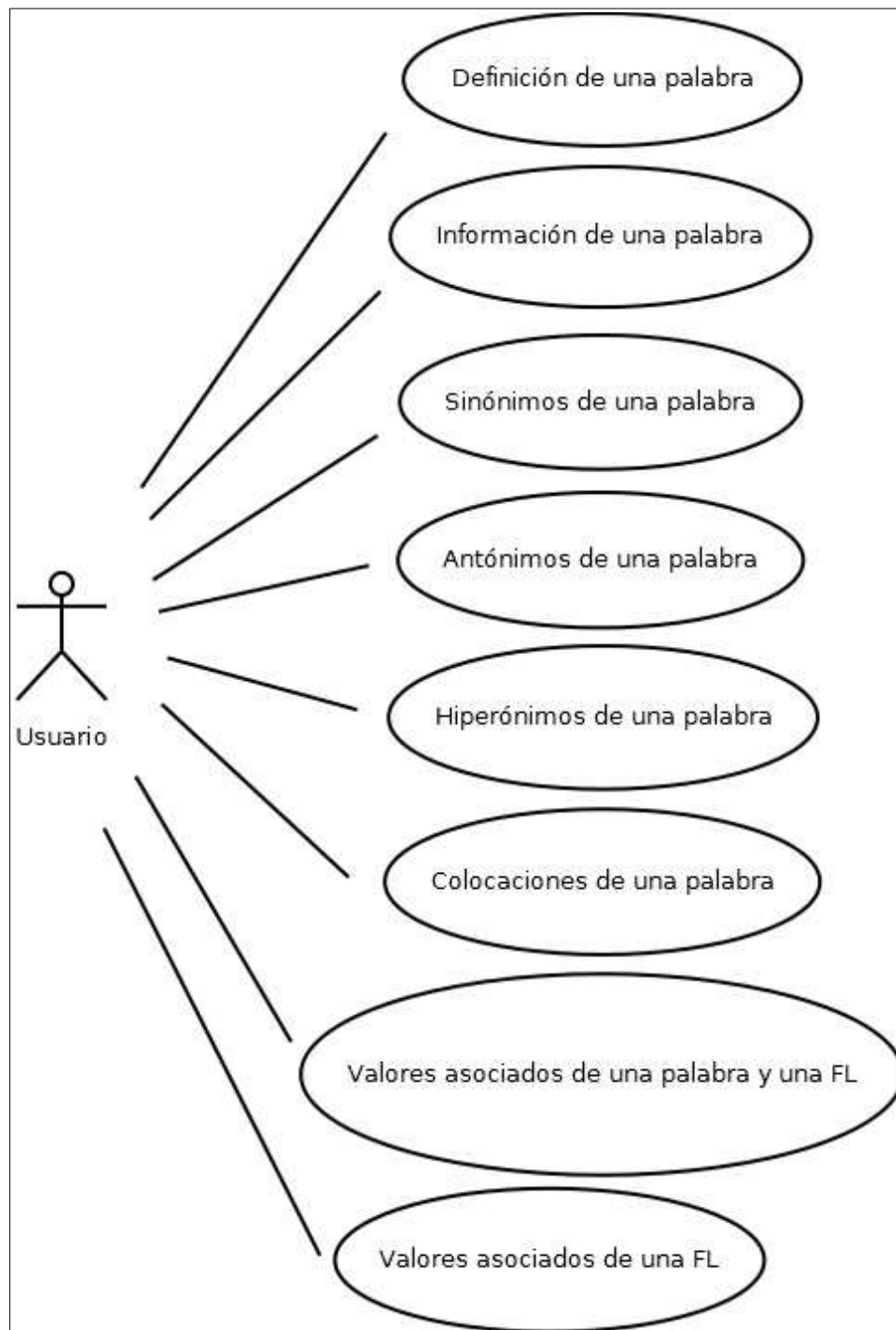


*Figura 6.1 -Diagrama de uso del servicio*

Los actores son un usuario (por ejemplo un lingüista) o una aplicación externa (por ejemplo una página web de traducción de texto). Ambos actores hacen uso del sistema, que es el servicio desarrollado.

Se distingue entre casos de uso en los que el iniciador del caso de uso es un usuario y los casos de uso en los que ese rol lo cumple una aplicación externa.

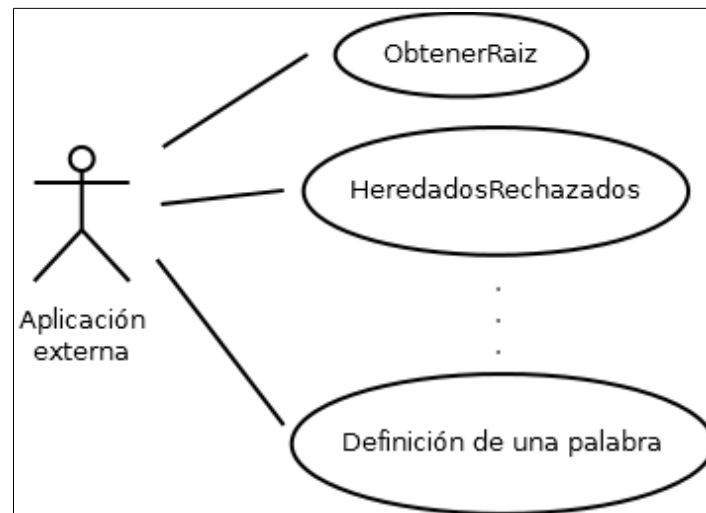
A continuación en la figura 6.2 se muestran los casos de uso en los que el iniciador es un usuario.



*Figura 6.2 -Diagrama de casos de uso iniciados por el usuario*

Hay ocho casos de uso, uno para cada una de las operaciones que se ofrecerán al usuario a través del interfaz.

En la figura 6.3 se muestran los casos de uso en los que el iniciador de la operación es una aplicación externa.



*Figura 6.3 -Diagrama de casos de uso iniciados por una aplicación externa*

En este caso, cada caso de uso se corresponde con una operación del servicio que puede ser invocada por un sistema externo (es importante tener en cuenta que estas operaciones incluyen las que un usuario puede ejecutar, ya que estas operaciones también las puede ejecutar una aplicación externa). Dado que hay un total de 70 casos de uso, se muestran sólo algunos de los casos de uso a modo de ejemplo.



## 6.2. Diagrama de clases

El diagrama de clases del sistema es el mostrado en la figura 6.4

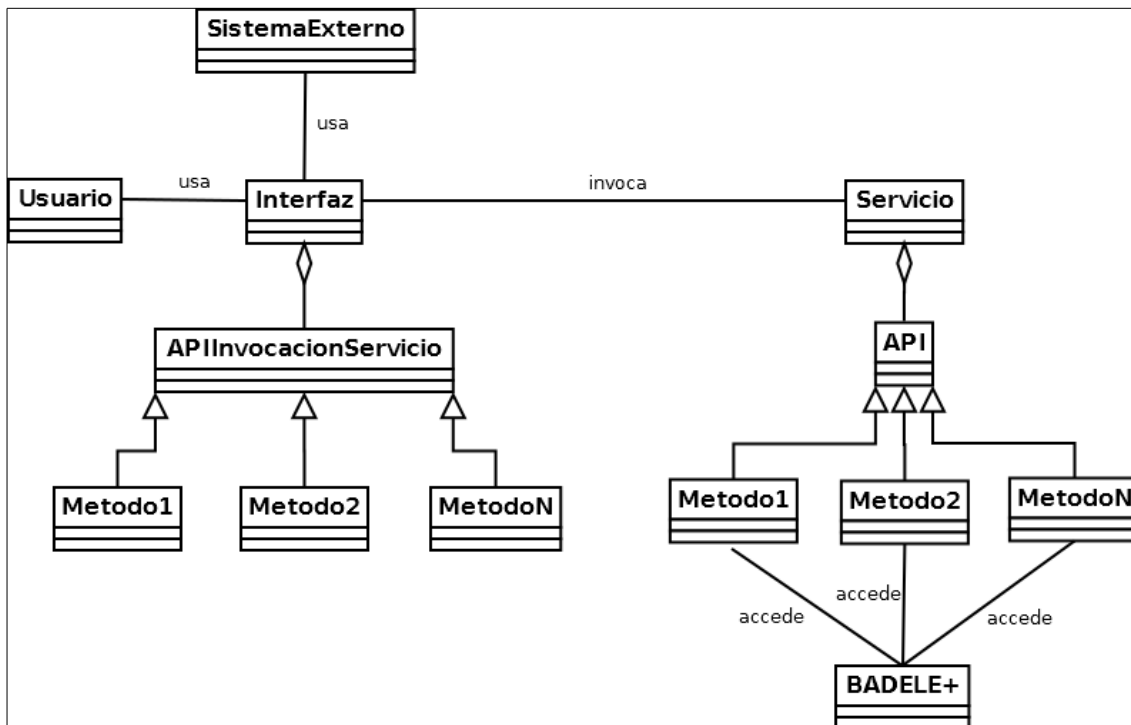


Figura 6.4 -Diagrama de clases

Un usuario o un sistema externo usan el interfaz, el cual contiene por un lado la representación visual del sistema y por otro los métodos de invocación de la funcionalidad disponible en el servicio (lo que se denomina APIInvocacionServicio, compuesto por métodos que invocan a su homólogo en el servicio). Por tanto, a través del interfaz se invoca al servicio, el cual contiene un API, formado por N métodos, que se apoyan en el acceso a la información contenida en BADELE+.

## 6.3. Diagramas de secuencia

Dado el gran número de operaciones, concretamente 91, y la similitud entre todas ellas, se van a mostrar tres a modo de ejemplo:

- Una operación que se puede ejecutar desde el interfaz web: *ObtenerColocaciones*.
- Una operación que no se puede ejecutar desde el interfaz web, pero sí con el API incluido en el mismo, y que es una consulta de información por parte de un sistema externo: *ObtenerRaiz*.
- Una operación que no se puede ejecutar desde el interfaz web, pero sí con el API incluido en el mismo, y que es una modificación de información de la base de datos: *InsertarDefinicion*.

### 6.3.1 ObtenerColocaciones

El diagrama de secuencia de *ObtenerColocaciones* se muestra en la figura 6.5.

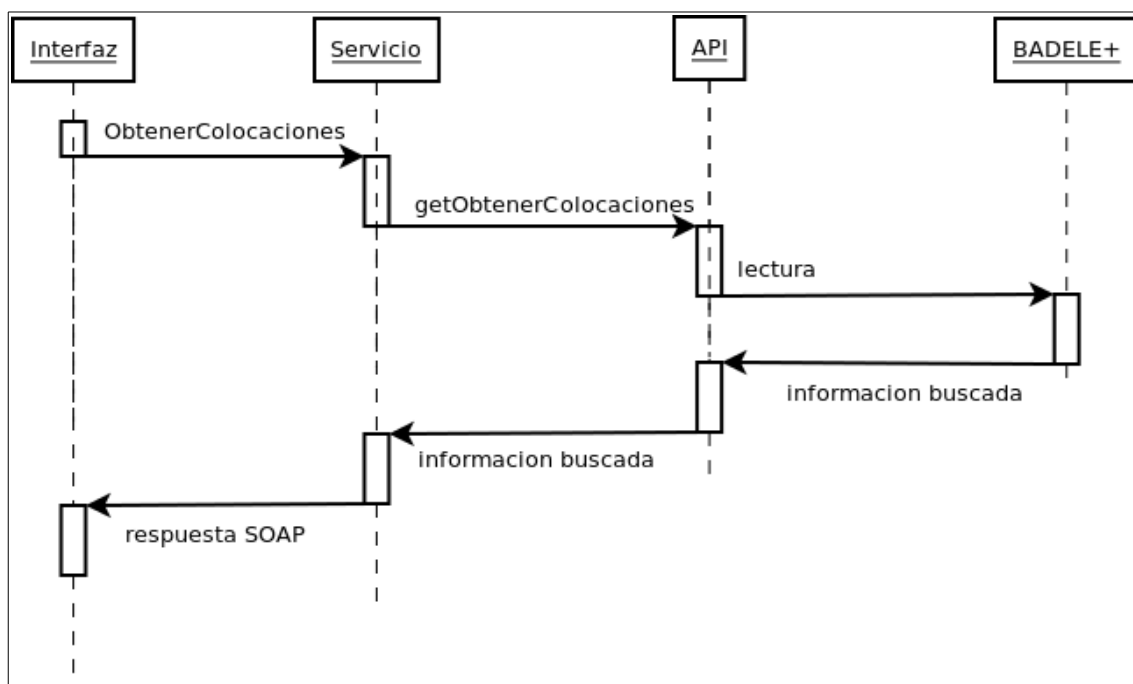


Figura 6.5 -Diagrama de secuencia *ObtenerColocaciones*

Los actores que aparecen son:

- Interfaz: en concreto, es el interfaz web, el cual a su vez hace uso del API de invocación del servicio.
- Servicio: el elemento que sirve las peticiones que se le realizan.
- API: conjunto de métodos que permiten obtener la información deseada.
- BADELE+: servidor de base de datos que contiene la información de BADELE+

A través del interfaz web se invoca al servicio indicando qué método del mismo se quiere ejecutar. Este servicio a su vez utiliza un API para, mediante el acceso a la base de datos a la base de datos, obtener la información buscada. Esta información es devuelta al servicio para que éste la retorne al interfaz (en forma de respuesta SOAP o similar).

### 6.3.2 ObtenerRaiz

El diagrama de secuencia de *ObtenerRaiz* es el mostrado en la figura 6.6.

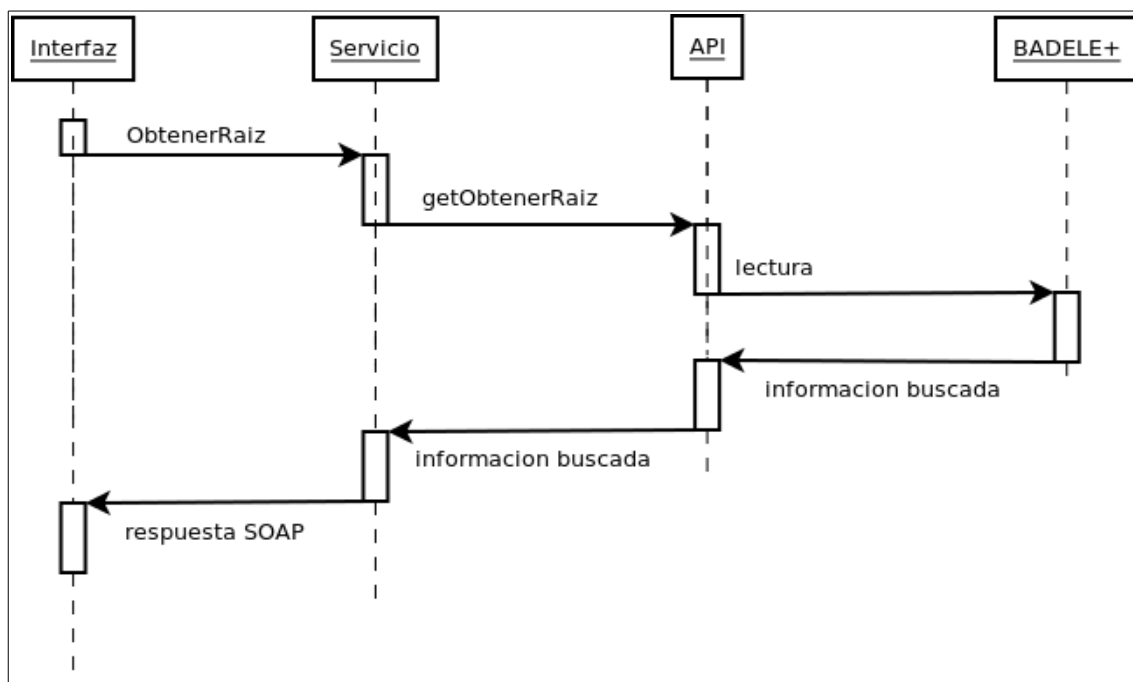


Figura 6.6 -Diagrama de secuencia *ObtenerRaiz*

Los actores que aparecen son:

- Interfaz: en concreto, es el API de invocación del servicio.
- Servicio: el elemento que sirve las peticiones que se le realizan.
- API: conjunto de métodos que permiten obtener la información deseada.
- BADELE+: servidor de base de datos que contiene la información de BADELE+

A través del API distribuída, que como se comentó anteriormente forma parte del interfaz, se invoca al servicio. En esa invocación se indica qué método del mismo se quiere ejecutar. Este servicio a su vez utiliza un API para, mediante el acceso a la base de datos, obtener la información buscada. Esta información obtenida es devuelta al servicio para que éste la retorne al interfaz (en forma de respuesta SOAP o similar).

### 6.3.3 InsertarDefinicion

El diagrama de secuencia de *InsertarDefinicion* se muestra en la figura 6.7.

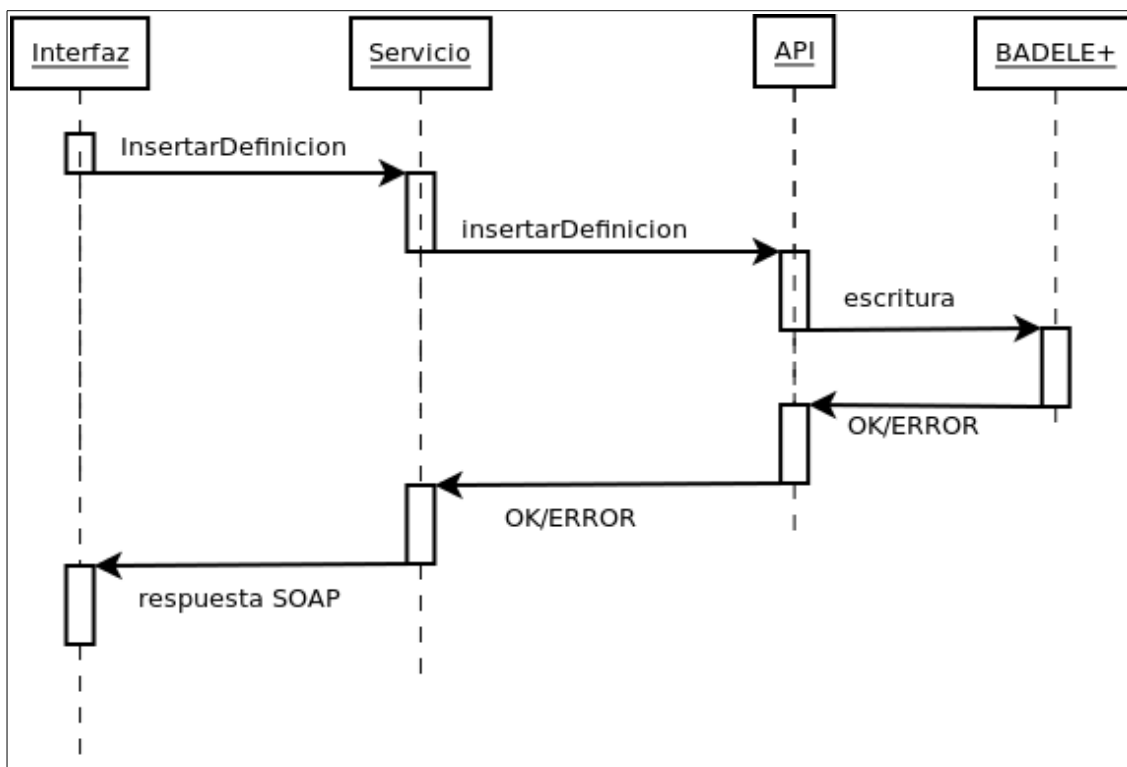


Figura 6.7 -Diagrama de secuencia *InsertarDefinicion*

Los actores que aparecen son:

- Interfaz: en concreto, es el API de invocación del servicio.
- Servicio: el elemento que sirve las peticiones que se le realizan.
- API: conjunto de métodos que permiten realizar los cambios deseados en la información.
- BADELE+: servidor de base de datos que contiene la información de BADELE+

A través del interfaz web se invoca al servicio indicando qué método del mismo se quiere ejecutar. Este servicio a su vez utiliza un API para, accediendo a la base de datos, insertar la información enviada desde el interfaz; la única respuesta que se obtiene es si la operación se ha realizado correctamente o si se ha producido algún error.

## 6.4. Implementación: organización y tecnologías

A continuación se ofrece un primer bosquejo del diseño del servicio, que no pretende ser más que la introducción a la arquitectura que se utiliza en el servicio. Este diseño se basa en una arquitectura de capas mostrada en la figura 6.8.

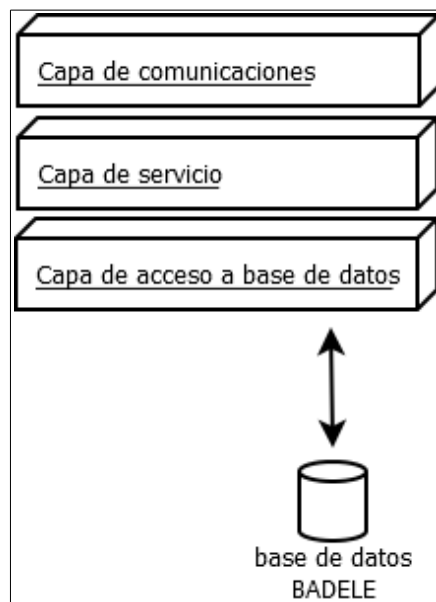


Figura 6.8 – Arquitectura del servicio

**Capa de acceso a base de datos.** Se trata del componente software encargado de interactuar con la base de datos, basado en un *pool* de conexiones JDBC abiertas con la base de datos utilizando el *driver* JDBC de MySQL, tal y como se muestra en la figura 6.9.

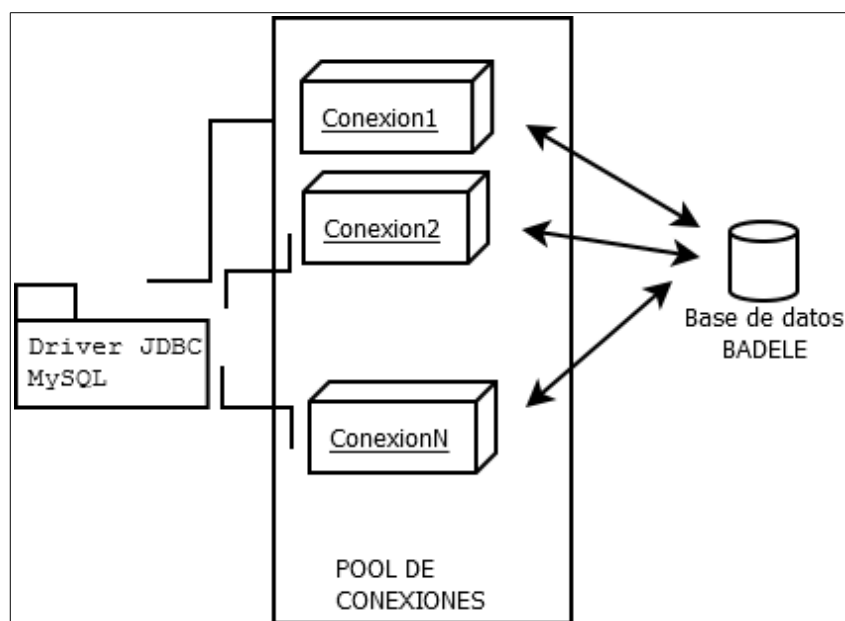


Figura 6.9 – Capa de acceso a base de datos.

Durante la ejecución del software el *pool* de conexiones se encarga de mantener N conexiones disponibles para ser utilizadas por el resto de capas de software; cada una de estas conexiones estará ya abierta con la base de datos, de manera que cuando se necesite su uso, no será necesario abrir la conexión, y así se mejora enormemente el rendimiento global del sistema.

Cada una de estas conexiones se abre con la base de datos a través de un *driver* JDBC de clase 4 específico de MySQL. El que sea de clase 4 quiere decir, básicamente, que está realizado íntegramente en Java y no necesita ningún componente nativo, lo que redundará en la portabilidad del software.

La capa de acceso a base de datos ofrece una serie de métodos que permiten el acceso a los objetos de la base de datos de dos maneras diferentes:

- Mediante el uso de sentencias SQL estándar.
- Mediante una abstracción de los objetos de base de datos (más concretamente las tablas del mismo) en forma de clases que representen la estructura de estas tablas.

**Capa de servicio.** Esta capa contendrá la lógica del servicio, esto es, los objetos encargados de “comprender” la entrada recibida de la capa de comunicaciones y realizar las operaciones necesarias para el desarrollo de la operación requerida (lo que probablemente incluirá interactuar con la capa de acceso a base de datos) para finalmente solicitar a la capa de comunicaciones el envío de alguna clase de mensaje al solicitante de la operación.

**Capa de comunicaciones.** La capa de comunicaciones se encarga de manejar la entrada y salida de información al servicio. Esta capa ofrece al exterior una serie de métodos que pueden ser invocados por cualquier cliente (de manera controlada) y que a su vez ofrecen una respuesta en un determinado formato.

Las comunicaciones entre el sistema y el API distribuída se realizan utilizando SOAP; esta tecnología se basa en peticiones POST sobre HTTP en las que se incluye un XML con un determinado formato; a su vez, la respuesta se realiza dentro de la propia operación POST con otro XML que contiene la información devuelta. Esto se puede ver en la figura 6.10.

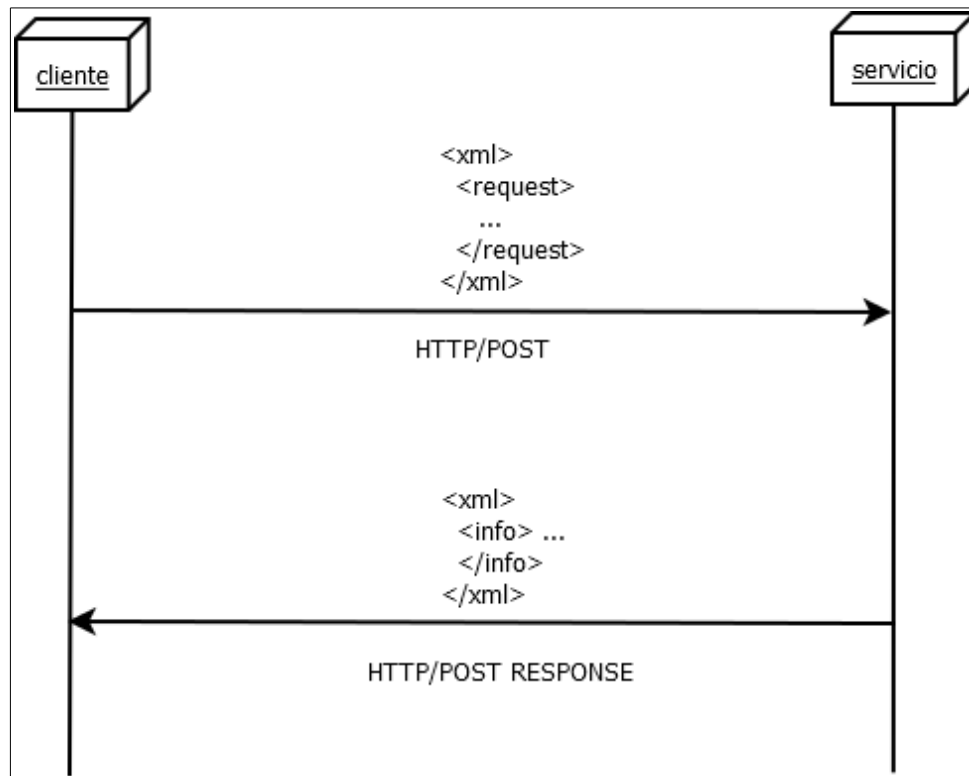


Figura 6.10 – Diagrama de comunicaciones

## 6.5. Desarrollo de la capa de acceso a base de datos

Dada su especial importancia en este proyecto, a continuación se desarrolla más en profundidad el funcionamiento de la capa de acceso a base de datos. El acceso a base de datos se basa en JDBC (*Java Database Connectivity*) que es una especificación basada en ODBC (*Open Database Connectivity*) formada por un protocolo y unos API's enfocados en la conexión de aplicaciones Java con una base de datos que lo admita. JDBC ofrece las ventajas inherentes de Java, esto es, independencia del sistema operativo sobre el que se ejecute y una teórica independencia también del gestor de base de datos con el que se trabaje (si bien esto no es exactamente así, los detalles técnicos no son relevantes en este caso). Otra de las ventajas de JDBC es la sencillez que tiene en el desarrollo, ya que los fabricantes de las bases de datos ofrecen librerías que implementan esta funcionalidad JDBC y que ocultan la mayor parte de los detalles, lo que facilita que el código que debe desarrollarse es casi siempre igual.

Por encima de la conexión JDBC se sitúa un *pool* de conexiones propio. Para utilizar este *pool* de conexiones, la primera operación que se debe realizar es cargar la librería JDBC de conexión al gestor de base de datos (específica de cada gestor) en el espacio de memoria de la máquina virtual Java que está ejecutando el programa. Una vez cargado el *driver*, debe iniciársele con los parámetros propios del mismo (como URL de conexión, *timeouts*, etc.).

Una vez cargado el *driver*, se debe abrir una conexión con la base de datos. Esta conexión supone un intercambio de información entre el *driver* JDBC y el gestor de base de datos específico del protocolo JDBC que permanece oculto para el desarrollador. Una vez abierta esa conexión, se realizan las operaciones deseadas y después de realizarlas, se libera esta conexión de base de datos



(esto se debe realizar forzosamente para liberar recursos de la base de datos, ya que normalmente el número de conexiones simultáneas que un gestor de base de datos acepta es limitado).

Estas operaciones suponen un trabajo que habría que realizar en cada una de las ejecuciones que impliquen acceso a base de datos; para evitar realizar algunas de ellas una y otra vez, se utilizan *pools* de conexiones a base de datos. Estos *pools* se pueden ver como almacenes de conexiones ya abiertas a base de datos, listas para ser utilizadas por el resto de componentes de la aplicación, de manera que se utilizan y después se retornan a ese almacén para que puedan volver a ser utilizadas; es este esquema de trabajo el que se utiliza en la implementación del sistema.

## 6.6. Diagrama de clases de la implementación

En la figura 6.11 se muestra el diagrama de clases de la implementación:

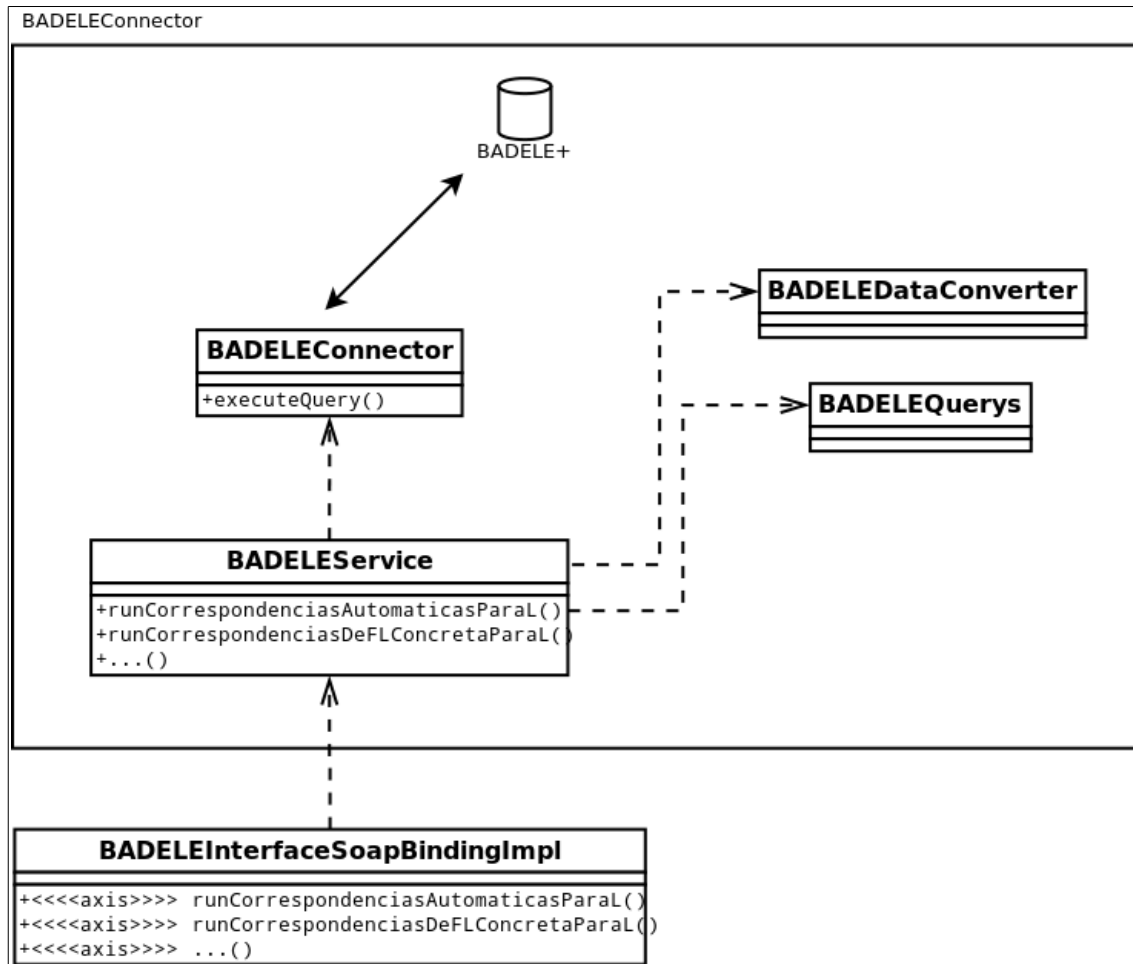


Figura 6.11 – Diagrama de clases del servicio

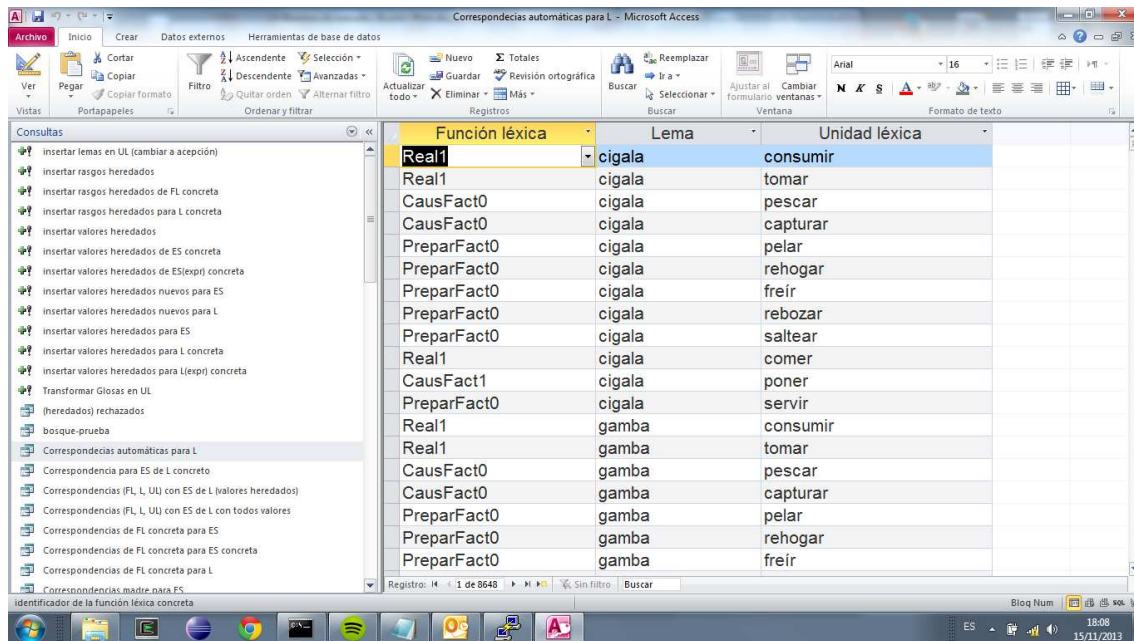
Estas son las clases principales del servicio, si bien hay otras auxiliares que se listarán posteriormente. De éstas, la clase **BADELEInterfaceSoapBindingImpl** es la que implementa los métodos del servicio que podrán ser invocados por un cliente remoto. Cada una de las operaciones mostradas en los diagramas de secuencia se corresponden con un método en esta clase. Cabe destacar que la estructura inicial de la clase (con los métodos vacíos) se obtiene utilizando las librerías de Axis.

La clase **BADELEInterfaceSoapBindingImpl** en cada uno de sus métodos realiza una serie de operaciones para, después, invocar a su método análogo de la clase **BADELEService**. Esta clase contiene un método por cada método de la clase **BADELEInterfaceSoapBindingImpl**; y cada uno de estos métodos es el que realiza las operaciones del servicio. Con esta estructura, se consigue separar la parte pública del servicio (los métodos de **BADELEInterfaceSoapBindingImpl**) de la parte privada, o sea la manera de funcionar interna, cuyo interfaz con la parte pública es **BADELEService**.

Esencialmente, en el funcionamiento de cada método de **BADELEService** se realiza una o varias llamadas a la clase **BADELEQuerys**, que contiene todas las consultas SQL que se realizan contra la base de datos. También utiliza la clase **BADELEDataConverter** que realiza una transformación entre datos de índices y datos relacionados con esos índices (esto se explica más adelante, en la sección 6.7). Finalmente, para ejecutar las consultas se invoca al método *executeQuery* de la clase **BADELEConnector** que es capaz de conectarse con la base de datos, ejecutar una consulta y devolver los resultados.

## 6.7. Conversión de datos

Las consultas originales se basan en una funcionalidad de Access que resulta bastante útil pero no es estándar. Consiste en mostrar en los resultados de una consulta como clave ajena entre tablas los campos de la tabla referenciada, en vez del índice de la tabla (que es el campo realmente referenciado). De manera, por ejemplo, que la consulta *CorrespondenciasAutomáticasparaL* devuelve en Access lo reflejado en la figura 6.12.



Función léxica	Lema	Unidad léxica
Real1	cigala	consumir
Real1	cigala	tomar
CausFact0	cigala	pescar
CausFact0	cigala	capturar
PreparFact0	cigala	pelar
PreparFact0	cigala	rehogar
PreparFact0	cigala	freír
PreparFact0	cigala	rebozar
PreparFact0	cigala	saltear
Real1	cigala	comer
CausFact1	cigala	poner
PreparFact0	cigala	servir
Real1	gamba	consumir
Real1	gamba	tomar
CausFact0	gamba	pescar
CausFact0	gamba	capturar
PreparFact0	gamba	pelar
PreparFact0	gamba	rehogar
PreparFact0	gamba	freír

Figura 6.12 – Resultado de una consulta en MSAccess

Sin embargo, esta misma consulta lanzada contra otro gestor de base de datos (en este caso MySQL pero lo mismo ocurriría con otro cualquiera) devuelve los resultados mostrados en la figura 6.13.

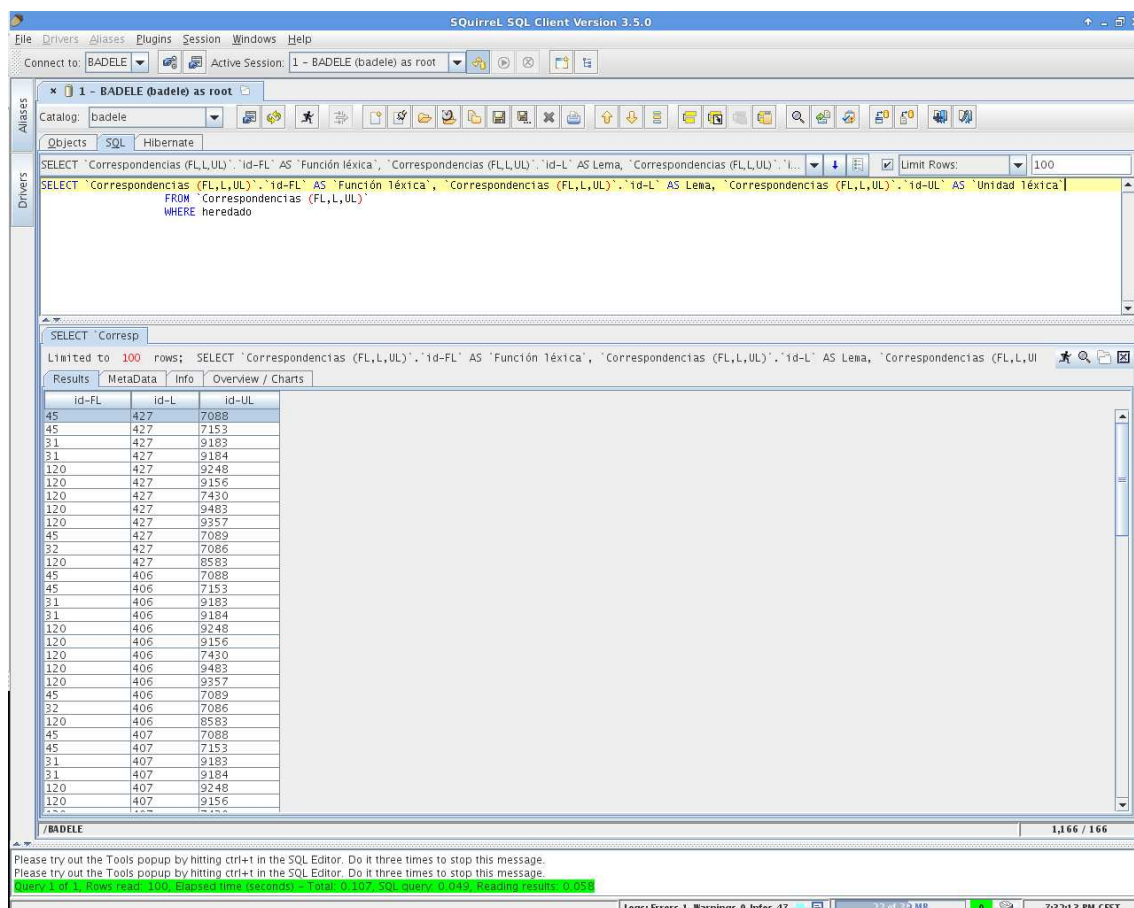


Figura 6.13 – Resultado de una consulta en MySQL

Como se puede observar, los resultados obtenidos son diferentes. El motivo de que sean diferentes es que Access, para facilitar su uso por parte de un usuario, convierte las claves de los registros referenciados en un campo de texto de las consultas, de manera que sea más sencillo para un usuario leer los resultados obtenidos. Para resolver esta diferencia sin que haya que modificar las propias consultas, lo que se hace es pasar los resultados por un filtro, implementado en la clase **BADELEDataConverter**, que lo que hace es buscar en determinadas tablas en base a los nombres de los campos. Por ejemplo, si una consulta retorna un campo con nombre id-ES, esta clase se encargará de acceder a la tabla 'Etiquetas Semánticas' para obtener el valor de la columna 'expresión' asociada al índice retornado en la consulta.

## 6.8. Interfaz

El interfaz web desarrollado como medio de acceso a la funcionalidad del servicio se basa en el patrón de arquitectura MVC. Este patrón divide las aplicaciones en tres capas de software, cada una de las cuales cumplen una función específica:

- **Modelo.** Es la capa encargada de acceder a base de datos, gestionando la conexión con la misma y las operaciones de lectura y escritura que se realizan.
- **Vista.** Es la capa que presenta la información a los usuarios, ya sea por pantalla, a través de una impresora o cualquier otro método de interacción hombre-máquina.
- **Controlador.** Es la capa intermedia entre las dos anteriores, la que contiene la lógica de la aplicación y conecta el resto de capas.

A continuación se detallan las tres capas del interfaz, que se ha denominado BADELEWeb. En las próximas secciones la expresión 'interfaz' se refiere a la aplicación web desarrollada con el fin de ofrecer a los usuarios un acceso sencillo y eficaz a la funcionalidad ofrecida por el servicio web descrito en las secciones anteriores.

### 6.8.1. Vista

La vista del interfaz se basa en HTML y Javascript; la pantalla inicial del interfaz es la mostrada en la figura 6.14.

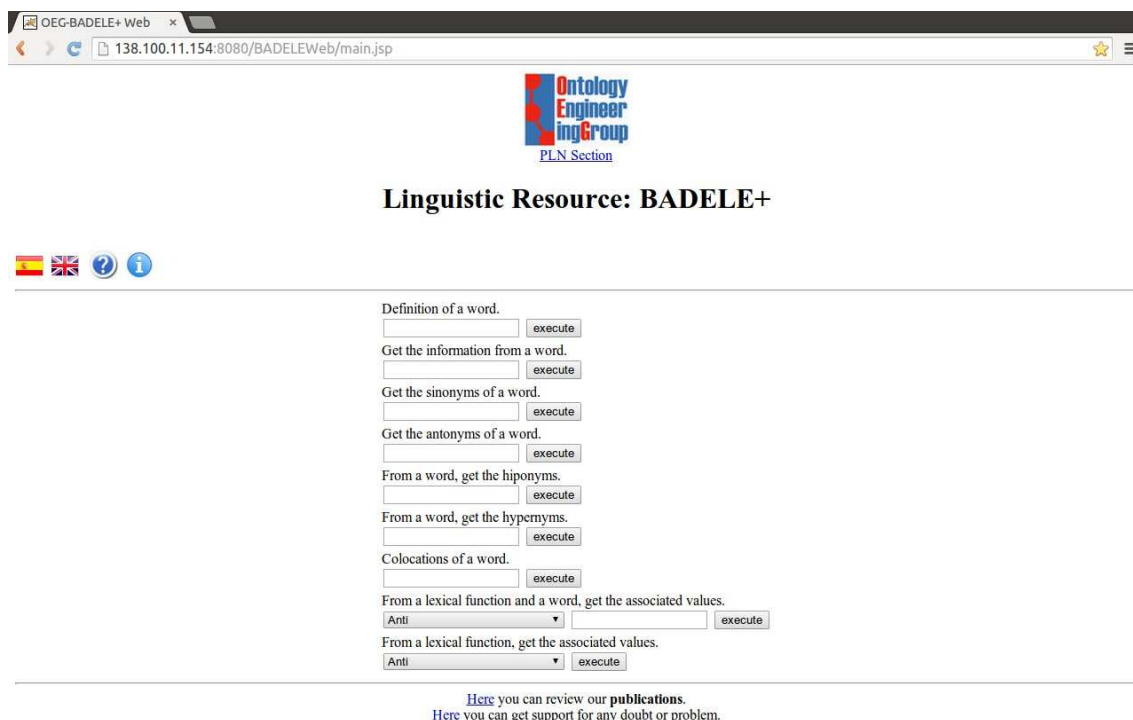


Figura 6.14 – Pantalla inicial del interfaz

En esta pantalla aparecen, en la parte superior, cuatro iconos que, izquierda a derecha, sirven para lo siguiente:

- La bandera española sirve para cambiar el idioma de la interfaz al español.
- La bandera británica sirve para cambiar el idioma de la interfaz al inglés.
- El símbolo de interrogación abre un documento de ayuda, aunque actualmente ese documento no está disponible.
- El símbolo con la letra 'i' abre un listado con las funciones léxicas dadas de alta en el sistema (esto se amplía en el apartado de características adicionales de la web).

Además, en la parte de abajo hay dos enlaces, uno de ellos a publicaciones del departamento y otro que contiene una dirección de correo electrónico para soporte y ayuda.

Al ejecutar una consulta se presentan los resultados en pantalla y se ofrece la opción de exportarlos a Excel, como se puede ver en la figura 6.15.

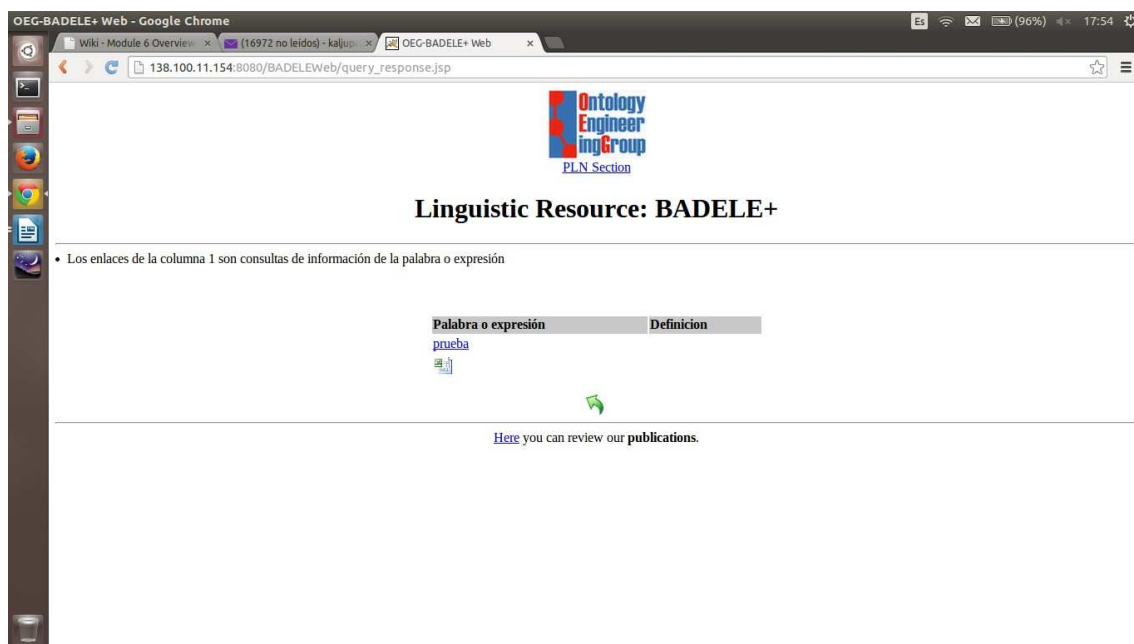


Figura 6.15 – Resultados de una consulta

Todo el resto del interfaz (cuyo funcionamiento se puede ver en detalle en la ayuda en línea) funciona con una filosofía similar.

### 6.8.2. Accesibilidad de la interfaz

La interfaz se ha desarrollado procurando mantener un diseño claro y sencillo. Además, se ha utilizado únicamente HTML estándar. Estos dos factores permiten que el acceso a la interfaz no está restringido a un PC. En la figura 6.16 se muestra la misma pantalla mostrada en la figura 6.14 accediendo a través de un dispositivo móvil.

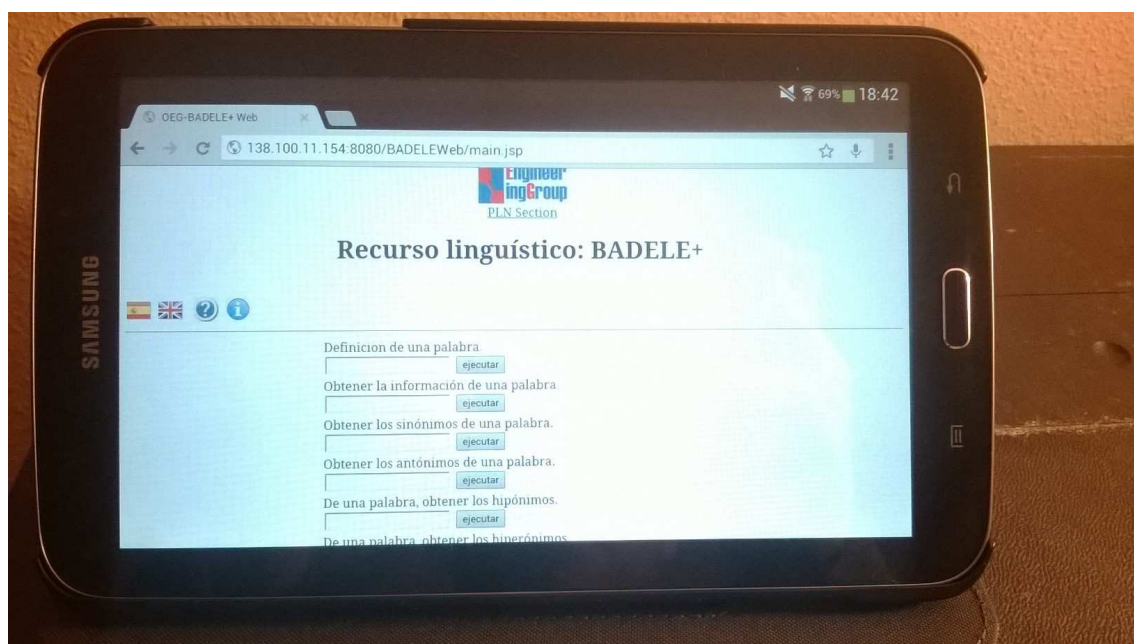


Figura 6.16 – Interfaz en un dispositivo portátil



Ésta es la misma pantalla de la figura 6.14 a la que se accede a través de una *tablet* con Android 4.1 y navegador Dolphin, esto es, un equipo totalmente alejado de un PC. En general, se puede decir que esta interfaz se puede utilizar a través de *tablets*, móviles y, en general, cualquier dispositivo capaz de procesar HTML y Javascript.

### 6.8.3. Modelo

En el caso de esta interfaz, el modelo no es un modelo de base de datos; en su lugar, el modelo estaría formado por el conjunto de funciones disponibles en el servicio del que se alimenta. Sin embargo, no todas las operaciones están disponibles en la interfaz, puesto que se ha considerado que la gran mayoría no resultan de interés para un usuario humano (que es quien accede al interfaz web). En concreto, las operaciones disponibles son las siguientes:

- Obtener la definición de una palabra.
- Obtener la información de una palabra.
- Obtener los antónimos de una palabra.
- Obtener los hipónimos de una palabra.
- Obtener los hiperónimos de una palabra.
- Obtener las colocaciones de una palabra.
- Obtener los valores asociados a una función léxica y a una palabra.
- Obtener los valores asociados a una función léxica.

Para acceder a estas operaciones, se utiliza un cliente generado mediante la librería Axis; a continuación, en la figura 6.17, se puede ver un modelo de las clases generadas.

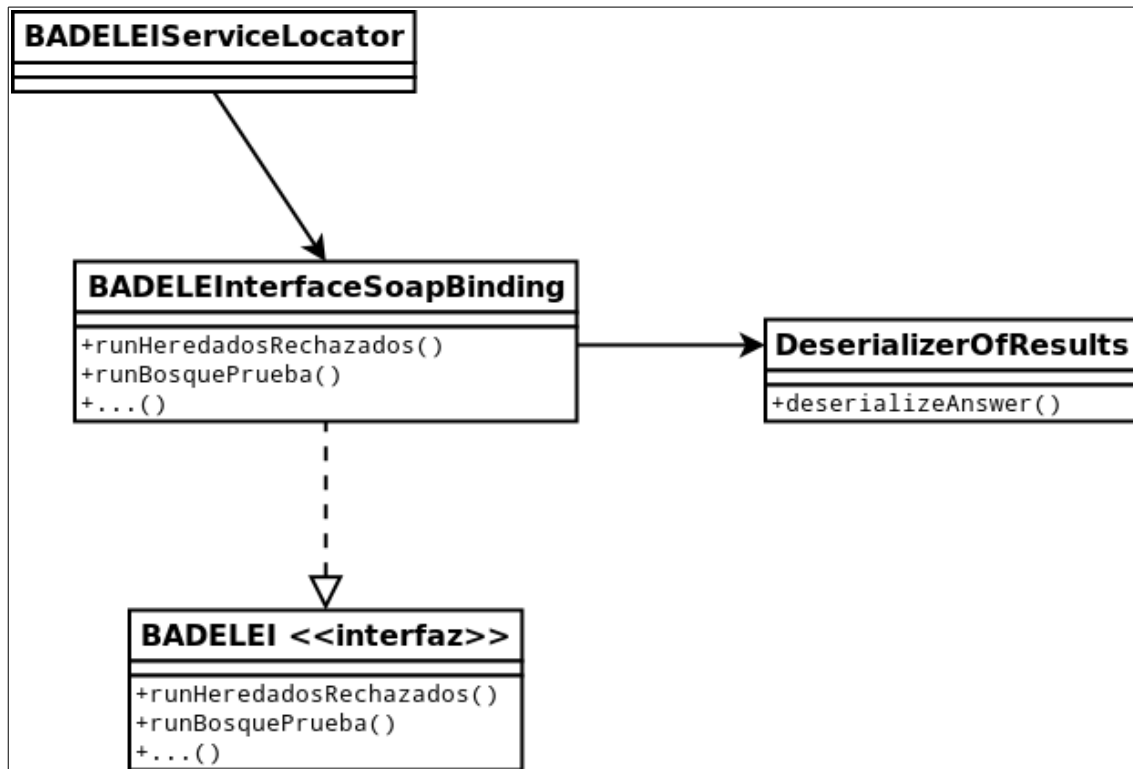


Figura 6.17 – Diagrama de clases del interfaz

**BADELEI:** es la interfaz (como elemento propio de la programación orientada a objetos) que representa todos los métodos que se ofrecen en el servicio. Se obtiene a través del WSDL, que es un fichero XML que el servicio mantiene público y en el que se describen todas y cada una de las funcionalidades ofrecidas.

**BADELEIServiceLocator:** es la clase que realiza la conexión con el servicio. Esta clase contiene los métodos que interactúan con el servicio, esto es, que realizan el diálogo necesario entre la interfaz web y el servicio cada vez que se desea invocar alguna operación del servicio. Aunque estos métodos se generan automáticamente, se debe modificar esta clase para que pueda indicarse la URL del servicio, que cambiará al realizar distintas instalaciones.

**BADELEInterfaceSoapBinding:** es la clase que implementa las llamadas al servicio. Esencialmente, contiene un método por cada una de las operaciones disponibles del servicio. Estos métodos se encargan de generar la petición SOAP, enviarla e interpretar los resultados SOAP que se reciben.

**DeserializerOfResults:** para evitar problemas de compatibilidad entre sistemas en el envío y recepción de caracteres que estén fuera del juego ASCII, los resultados propios de cada método (por ejemplo, el contenido de una determinada tabla) se serializa como un objeto XML y se codifican los caracteres del contenido antes de ser enviados desde el servicio. Esta es la clase que en el cliente se encarga del proceso contrario, convirtiendo ese contenido en XML en contenido preparado para ser mostrado por la pantalla.

## 7. RESULTADOS

## 7.1 Resultados

La figura 7.1 ofrece una visión general del sistema completo desarrollado:

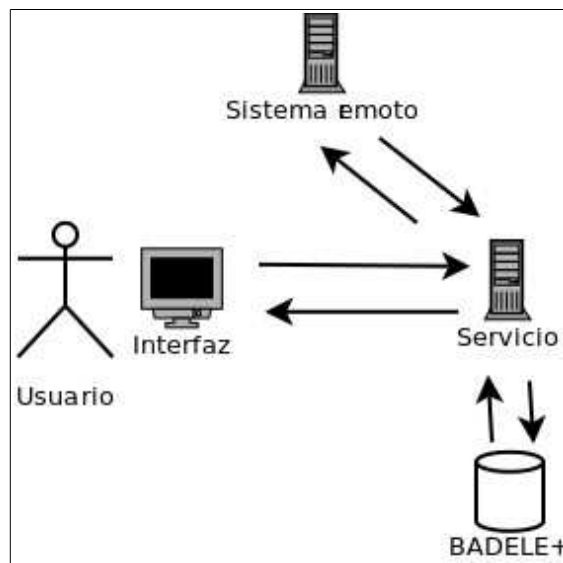


Figura 7.1 – Vista general del sistema

Los resultados han sido:

- La exportación de BADELE+ desde MSAccess a MySQL.
- El desarrollo de un servicio que ofrece funciones que permiten el acceso a la información contenida en BADELE+.
- El desarrollo de un interfaz web, capaz de ofrecer acceso a la parte de la funcionalidad ofrecida por el servicio que se ha considerado como útil para un usuario humano.

Con este desarrollo, los requisitos han sido satisfechos de la siguiente manera:

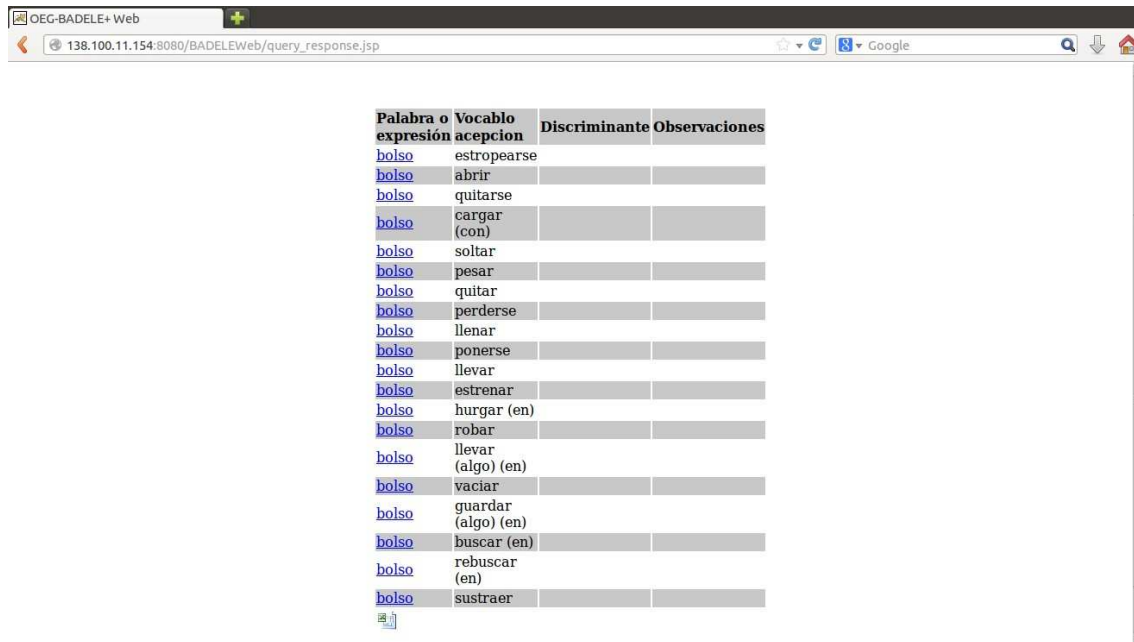
- Los requisitos del Fun #0001 a Fun #0010 han sido cubiertos mediante el desarrollo del servicio.
- Los requisitos Fun0011 y Fun0012, así como Int #0001 a Int #0004 han sido cubiertos mediante el interfaz.

Además el desarrollo se ha realizado respetando los requisitos Arq #0001, Arq #0002 y Arq #0003.

## 7.2 Experimentación

A continuación se muestran tres ejemplos de uso del proyecto desarrollado, tanto por usuarios como por otros sistemas. Así mismo, se da una muestra del rendimiento que el sistema es capaz de ofrecer.

**Ejemplo 1.** Un traductor necesita saber los verbos que se usan con el sustantivo bolso, para ello realiza la búsqueda de las colocaciones de la palabra 'bolso', obteniendo resultado de la figura 7.2:



Palabra o expresión	Vocablo acepcion	Discriminante	Observaciones
<a href="#">bolso</a>	estropearse		
<a href="#">bolso</a>	abrir		
<a href="#">bolso</a>	quitarse		
<a href="#">bolso</a>	cargar (con)		
<a href="#">bolso</a>	soltar		
<a href="#">bolso</a>	pesar		
<a href="#">bolso</a>	quitar		
<a href="#">bolso</a>	perderse		
<a href="#">bolso</a>	llenar		
<a href="#">bolso</a>	ponerse		
<a href="#">bolso</a>	llevar		
<a href="#">bolso</a>	estrenar		
<a href="#">bolso</a>	hurgar (en)		
<a href="#">bolso</a>	robar		
<a href="#">bolso</a>	llevar (algo) (en)		
<a href="#">bolso</a>	vaciarse		
<a href="#">bolso</a>	guardar (algo) (en)		
<a href="#">bolso</a>	buscar (en)		
<a href="#">bolso</a>	rebuscar (en)		
<a href="#">bolso</a>	sustraer		

Figura 7.2 – Resultados de la búsqueda de colocaciones

**Ejemplo 2.** Un estudiante de castellano necesita conocer a qué se refiere la palabra coche, para lo que utiliza la opción de obtener información de la palabra. Al hacerlo obtiene los resultados de la figura 7.3. Como puede verse, en este caso interfaz se ha seleccionado en inglés, que es una de las mejoras respecto a los sistemas existentes.



  
**Linguistic Resource: BADELE+**

---

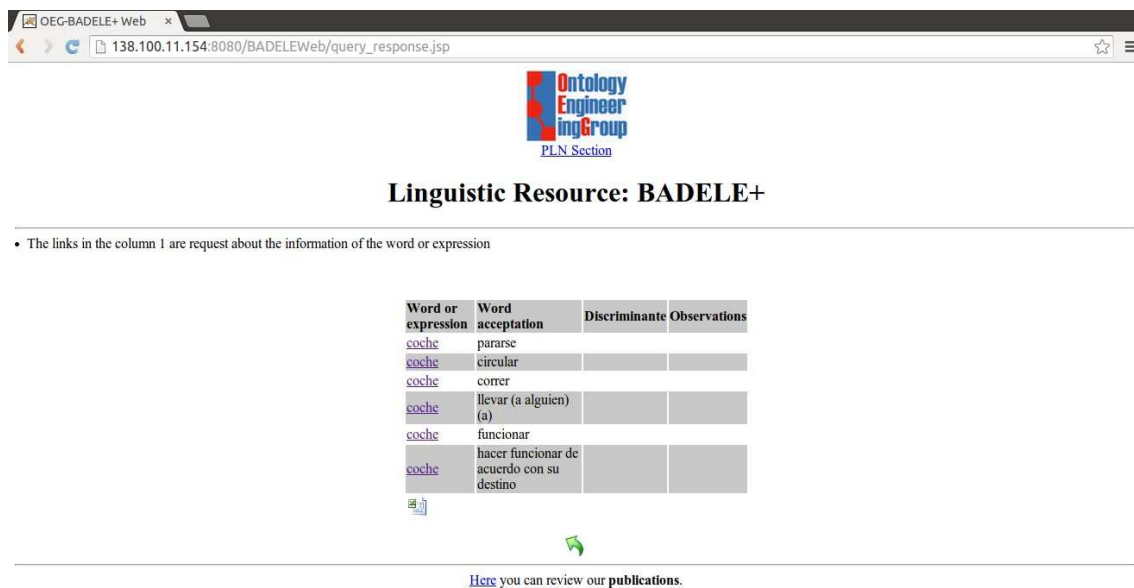
- The links in the column 1 are request about the collocations of the word or expression

Word or expression	Expression	Definition	Aception
<a href="#">coche</a>	Medio de transporte		coche

[Here](#) you can review our publications.

Figura 7.3 – Resultados de la consulta de información

Para conocer mejor el uso de esa palabra, el estudiante pincha sobre la misma, obteniendo las colocaciones de la palabra coche, tal y como se muestra en la figura 7.4. Esta capacidad de enlazar operaciones es otra mejora respecto a las aplicaciones existentes.



The screenshot shows a web browser window with the URL `138.100.11.154:8080/BADELEWeb/query_response.jsp`. The page features the logo of the 'Ontology Engineering Group' and the text 'PLN Section'. Below this, the title 'Linguistic Resource: BADELE+' is displayed. A note states: 'The links in the column 1 are request about the information of the word or expression'. A table with four columns is shown: 'Word or expression', 'Word acceptation', 'Discriminante', and 'Observations'. The table contains six rows of data for the word 'coche'. At the bottom, there is a link: 'Here you can review our publications.'

Word or expression	Word acceptation	Discriminante	Observations
<a href="#">coche</a>	pararse		
<a href="#">coche</a>	circular		
<a href="#">coche</a>	correr		
<a href="#">coche</a>	llevar (a alguien) (a)		
<a href="#">coche</a>	funcionar		
<a href="#">coche</a>	hacer funcionar de acuerdo con su destino		

Figura 7.4 – Colocaciones de la palabra coche

**Ejemplo 3.** Otros posibles usos incluyen la invocación por parte de otros sistemas de los métodos disponibles. Esta es una clara mejora respecto a los sistemas actualmente existentes, ya que ninguno de ellos ofrece esta posibilidad. Por ejemplo, un sistema que necesita obtener la palabra con sentido opuesto a 'alto' puede ejecutar el método del servicio *runGetValoresAsociados* obteniendo como resultado un XML que contiene la palabra 'bajo'; para ello, simplemente tendrá que ejecutar las siguientes líneas de código (lo mostrado es Java, pero puede utilizarse cualquier otro lenguaje de programación):

```
BADELEIServiceLocator locator = new BADELEIServiceLocator();
BADELEI interf = locator.getBADELEInterface();
ArrayList<ArrayList<ResultBean>> res = new
DeserializerOfResults().deserializeAnswer(interf.runGetValoresAsociados("userbad ele",
"passwdbadele", "anti", "alto"));
ResultBean r = (ResultBean)res.get(0).get(0);
System.out.println(r.getValue());
```

En la figura 7.5 se puede ver el entorno de desarrollo con un programa de prueba que se trata de este mismo código. Se puede ver que la operación invocada es *runGetValoresAsociados*. En la parte de abajo de la imagen, aparecen los resultados obtenidos al ejecutarlo, en concreto la palabra 'bajo'.

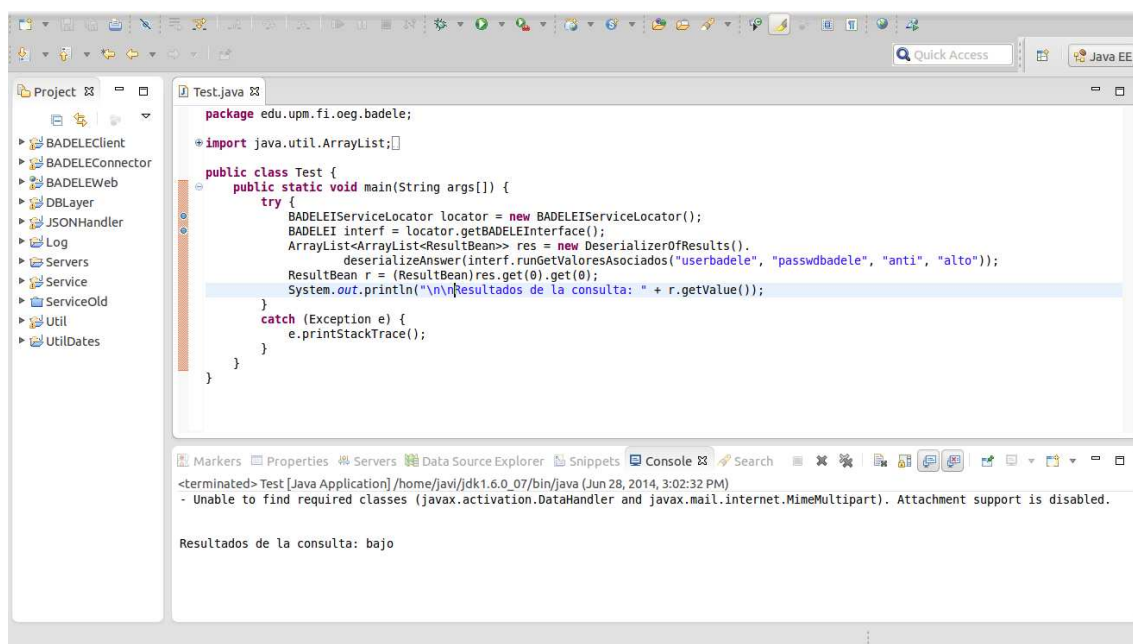


Figura 7.5 – Entorno de desarrollo con un programa de pruebas

**Rendimiento.** Como parte de las pruebas realizadas, se ha llevado a cabo una prueba de carga consistente en obtener los valores asociados a la palabra 'alto' (que es de hecho el ejemplo anterior), por parte de diez procesos concurrentes y un total de 1.000 consultas por cada proceso. Las consultas se realizan desde un ordenador portátil conectado a través de internet con el servicio, mediante un pequeño programa desarrollado en Java (similar, con algunas modificaciones, al mostrado en la figura 7.5). Se obtiene que el tiempo total necesario para realizar las 10.000 consultas es de 2 minutos y 3 segundos, lo que da una tasa de 49,5 consultas atendidas por el servicio por segundo.

Esta tasa de consultas atendidas por minuto es únicamente válida con el actual volumen de datos, siendo necesario repetir esta prueba en caso de que el volumen se incremente en gran medida.

Se han realizado otras pruebas, en concreto pruebas unitarias y funcionales, para comprobar que el funcionamiento tanto del servicio como de la interfaz web son correctos, obteniendo finalmente un resultado exitoso en todas ellas. No se considera de interés, sin embargo, listarlas en este documento, ya que no aportan información útil al mismo.

## 7.3 Características adicionales de usabilidad de la web

A lo largo del desarrollo, se han ido observando la conveniencia de añadir algunas funcionalidades adicionales que, si bien no estaban inicialmente recogidas en los requisitos, permiten una mejor experiencia de usuario. Algunas de estas características se han podido observar ya en el apartado anterior, si bien se considera interesante explicarlas todas estas características juntas, aun en el caso de que ya se hayan comentado. A continuación se describen estas funcionalidades:

**Enlaces entre resultados.** Al realizar una determinada consulta, los resultados que aparecen a partir de la misma pueden, en muchos casos, permitir ejecutar directamente otra consulta con el término seleccionado o con otros que aparezcan como parte de los resultados, simplemente pinchando con el ratón sobre el término que se quiera. Por ejemplo, ejecutando la consulta de hiperónimos de la palabra 'coche' se obtiene el resultado mostrado en la figura 7.7.



The screenshot shows a web browser window with the address bar displaying '138.100.11.154:8080/BADELEWeb/query\_response.jsp'. The page features the 'Ontology Engineering Group' logo and the text 'PLN Section'. Below this, the title 'Linguistic Resource: BADELE+' is centered. A list of links is provided: 'The links in the column 1 are request about the information of the word or expression' and 'The links in the column 2 are request about the information of the word or expression'. A table with five columns is shown: 'Word or expression', 'Expression', 'Glosa', 'Observations', and 'Inferencia semantica'. The first row contains the word 'coche' in the first column, 'Medio de transporte' in the second column, and 'true' in the fifth column. A small icon of a car is visible below the word 'coche'. At the bottom of the table, there is a link: 'Here you can review our publications.'

Word or expression	Expression	Glosa	Observations	Inferencia semantica
<a href="#">coche</a>	<a href="#">Medio de transporte</a>			true

Figura 7.7 – Resultados de la consulta de hiperónimos de la palabra 'coche'

Como puede observarse, tanto 'coche' como 'Medio de transporte' pueden ser pinchados con el ratón. En caso de hacerlo sobre 'coche' aparecerían los resultados mostrados en la figura 7.3. En caso de hacerlo sobre 'Medio de transporte', se ejecuta la consulta de información de 'Medio de transporte' exactamente igual que si se realizara desde la pantalla inicial. Esta consulta no tiene actualmente información en la base de datos, por lo que se obtiene una lista vacía, y por eso no se muestra la captura de pantalla.

**Ayuda de las funciones léxicas.** Dado el gran número de funciones léxicas, se ha incluido un apartado en el interfaz que ofrece un listado de las mismas ordenadas alfabéticamente y la descripción de cada una de ellas que esta contenida en la base de datos; esto se muestra en la figura 7.8.



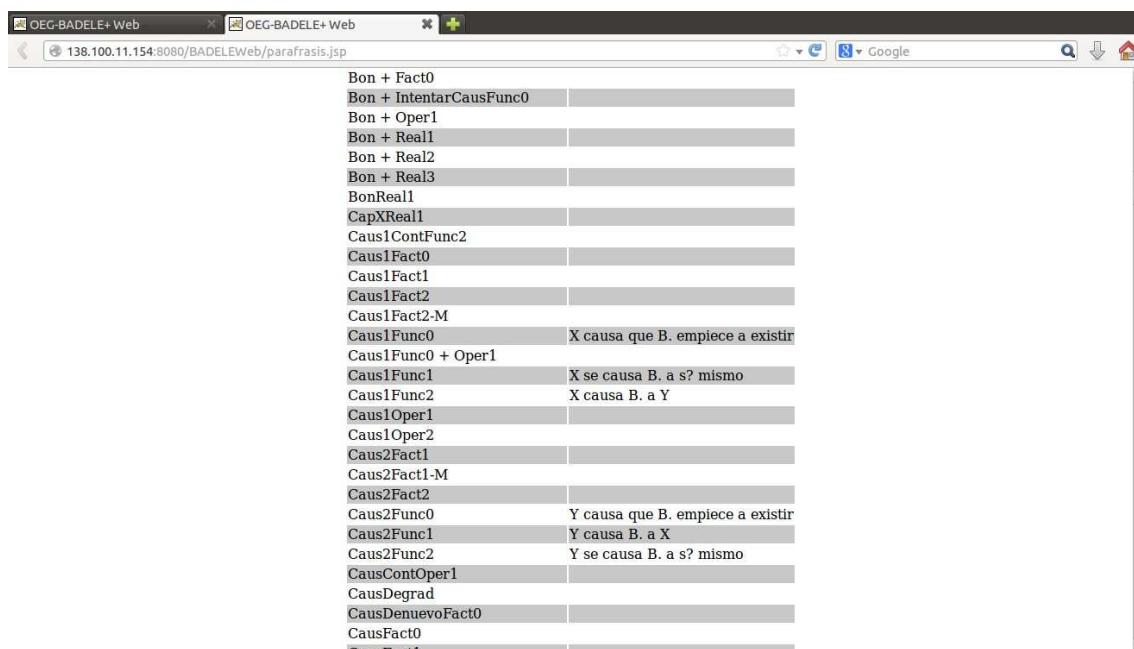


Figura 7.8 – Ayuda de funciones léxicas

Se trata de una ayuda sencilla, que actualmente no permite ninguna clase de búsqueda sobre ella, salvo la búsqueda texto que se puede realizar con el propio navegador web.

Además, en caso de realizar una consulta que incluya como resultado una función léxica, al pulsar sobre esa función se obtiene esa misma descripción, como por ejemplo en el caso de Caus1Func0 (figura 7.9)

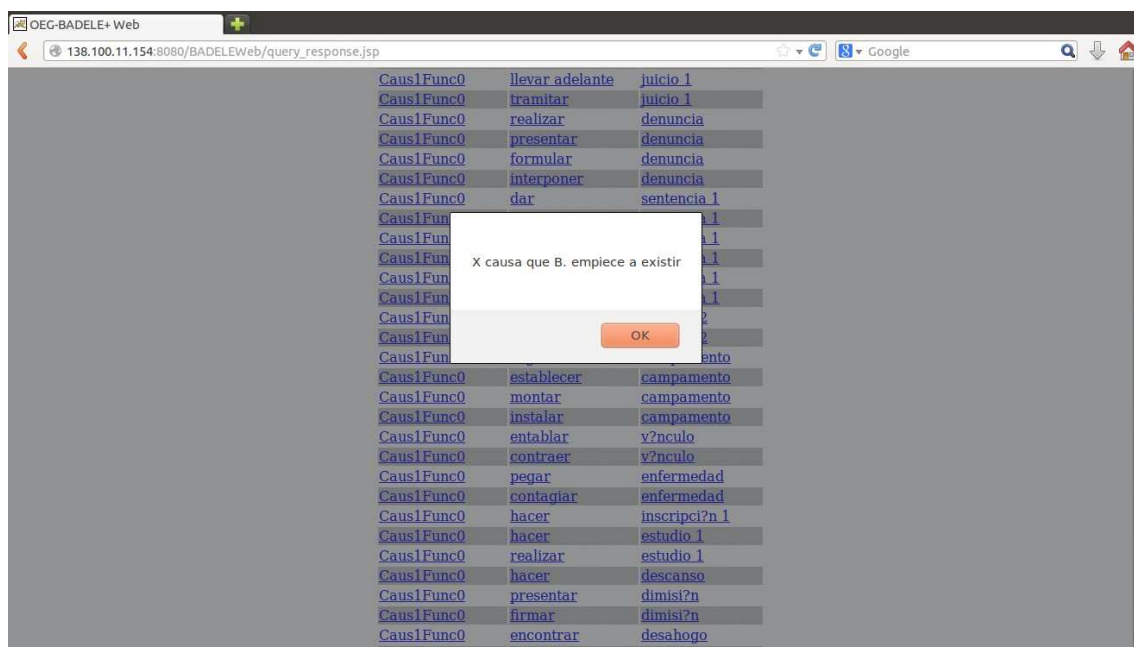


Figura 7.9 – Ayuda de una función en concreto

## 7.4 Comparativa con otras herramientas

A continuación se muestra una tabla similar a la mostrada en el apartado 2.4, a la que se añade una nueva columna con las características de BADELE+, a modo de resumen de las mejoras introducidas con respecto a las herramientas ya existentes.

Herramienta	DiCoInfo	Dicouèbe	DiCE	BADELEWeb
<b>Grupo desarrollador</b>	Universidad de Montreal	Laboratorio LaTTiCe	Grupo Dice (Universidad de La Coruña)	OEG (FI-UPM)
<b>Idiomas contenidos</b>	Frances, Inglés, Español	Frances	Español	Español
<b>Operaciones web</b>	Búsqueda de términos utilizando distintos filtros	Búsqueda de vocablos, funciones léxicas, ejemplos, etc	Consulta de colocaciones, significado de términos, ayuda a la redacción	Consultas de definiciones, sinónimos, etc (ver requisito Fun #0010)
<b>Operaciones remotas</b>	No tiene	No tiene	No tiene	Dispone de un interfaz para invocación por otros sistemas
<b>Librerías para uso remoto</b>	No tiene	No tiene	No tiene	Disponibles librerías para invocación por parte de sistemas remotos
<b>Enlace entre consultas</b>	Enlaces a términos relacionados	No tiene	Enlaces a colocaciones	Enlaces entre consultas del interfaz relacionadas
<b>Idiomas del interfaz</b>	Frances, Inglés, Español	Frances	Español	Español, Inglés
<b>Exportación de resultados</b>	No tiene	Permite exportar los resultados a un fichero de texto	No tiene	Permite exportar resultados a Excel
<b>Ayuda</b>	Contiene un manual de ayuda, así como un resumen, pero únicamente en frances	Contiene un manual únicamente en frances	Posee ayuda en línea en castellano	Posee un documento de ayuda, así como ayuda en línea sobre el significado de las funciones léxicas

Como ventajas de BADELEWeb frente al resto de sistemas se pueden destacar:

- El gran número de operaciones incluidas en el sistema (considerando las operaciones para trabajo entre máquinas, no visibles en el interfaz, pero que sí están listadas en el anexo I), mayor que las existentes en el resto de sistemas.
- La posibilidad de acceder a las operaciones por otros sistemas automáticos, existiendo incluso una librería que facilita enormemente ese acceso. Esta librería, desarrollada en java, es parte de la interfaz web, pero puede utilizarse como pieza separada para que otro software acceda los métodos del servicio sin tener que preocuparse de cómo funciona ese acceso.
- La interfaz ofrece facilidades como los enlaces entre resultados, la traducción a dos idiomas del interfaz y la exportación a un formato más manejable (MS Excel) que no están presentes en todos los sistemas.
- Posee una ayuda muy completa, gracias a la documentación del sistema y a la información sobre las funciones léxicas. Cabe indicar, sin embargo, que actualmente la documentación no está todavía disponible, aunque es de esperar que en poco tiempo se pueda poner accesible.



## 8. LINEAS FUTURAS

El desarrollo de este trabajo tiene como consecuencia la aparición de determinadas líneas de desarrollo relacionadas con el propio sistema obtenido.

Algunos de los trabajos que se proponen a raíz de este proyecto se exponen a continuación estructurados de la siguiente manera:

- Motivación de la mejora, esto es, el porqué del interés de esa mejora.
- Posible solución para llevar a cabo esa mejora.

## 8.1. Perfilado de usuarios

### **Motivación de la mejora**

El interfaz web ofrecido es único para todos los usuarios, sea cual sea el usuario. Sin embargo, la funcionalidad necesitada por un usuario dependerá, probablemente, del tipo de usuario. Por ejemplo, un traductor podría requerir una funcionalidad diferente a la requerida por un lingüista. Por tanto, sería una mejora interesante que el interfaz se personalizara en función del perfil del usuario que lo utilice.

### **Propuesta de solución**

Una posible solución sería la introducción de un usuario y contraseña en el interfaz, de manera que cada persona que necesite acceso se valide con sus datos, teniendo a su vez cada usuario asignado un perfil de usuario diferente. En ese caso, cada perfil tendría un idioma por defecto propio, así como unas funcionalidades ofrecidas personalizables por el propio usuario.

## 8.2. Interfaz para dispositivos móviles

### **Motivación de la mejora**

El interfaz desarrollado es mostrado correctamente en dispositivos móviles, tales como teléfonos o *tablets*. Sin embargo, al no estar específicamente orientado a ellos, puede resultar incómodo el uso del mismo en determinados dispositivos (sobre todo aquellos con pantallas más pequeñas). Además, se desaprovechan las características de interfaz específicas de la plataforma.

### **Propuesta de solución**

La solución sería el desarrollo de un interfaz diseñado específicamente para dispositivos móviles y la utilización de una u otra interfaz en función del dispositivo con que se acceda al mismo. Ésta es una solución muy común hoy en día y permitiría el acceso más cómodo a la herramienta desde cualquier dispositivo con conexión a internet.

### 8.3. Mejora de la entrada de información

#### **Motivación de la mejora**

Actualmente, la entrada de nueva información en BADELE+ no está automatizada, de manera que se debe realizar manualmente, lo que es un trabajo lento y poco eficiente.

#### **Propuesta de solución**

La solución propuesta se debe realizar en dos partes. Primero, se debe estudiar minuciosamente el tipo de información que actualmente se está incorporando y cómo se está llevando a cabo esa tarea, para en segundo lugar diseñar las herramientas software que permitan optimizar ese proceso.





## 9. BIBLIOGRAFÍA

La bibliografía utiliza el sistema de citas y referencias bibliográficas Harvard - Asociación Americana de Psicología (Harvard-APA). Este sistema puede consultarse en <http://apastyle.org/>.

BARRIOS, M.A. (2008) Tesis Doctoral Madrid.

CONCEPTO DE, DEFINICIÓN DE (2012) Definición de léxico.

Consultada el 15 de Agosto de 2012, de

<http://conceptodefinicion.de/lexico/>

DICOUÈBE (2013) Accueil — DiCouèbe : le DiCo en ligne - OLST

Obtenido el 20 de Junio de 2014, de

<http://olst.ling.umontreal.ca/dicouebe/>

DICE (2013) Pagina de bienvenida

Obtenido el 20 de Junio de 2014, de

<http://www.dicesp.com/paginas>

GIRALDO, J.J. (2012) La Teoría Sentido Texto, Introducción.

Obtenido el 15 de Agosto de 2012, de

<http://www.iula.upf.edu/materials/030313giraldo.pdf>

IEEE EXPLORE (2013) IEEE Sp 830, 98.

Obtenido el 20 de Septiembre de 2012, de

<https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?Punumber=5841>

L'HOMME (2009) DiCoInfo Dictionary of Computer Science and Internet Terms

Obtenido el 20 de Septiembre de 2012, de

[http://olst.ling.umontreal.ca/dicoinfo/DiCoInfo%20manual\\_Eng.pdf](http://olst.ling.umontreal.ca/dicoinfo/DiCoInfo%20manual_Eng.pdf)

MEL'CUK, IGOR, POLGUERE, ALAIN. (2007). Lexique actif du français. Bruxelles. Champs linguistiques.

WIKIPEDIA/UNIDAD LÉXICA (2013)

Consultada el 15 de Agosto de 2012, de

[http://es.wikipedia.org/wiki/Unidad\\_léxica](http://es.wikipedia.org/wiki/Unidad_léxica)

## ANEXO I – OPERACIONES DISPONIBLES EN EL SERVICIO

A continuación se pueden ver las operaciones disponibles en el sistema:

- Raíz
- Heredados Rechazados
- Bosque Prueba
- Correspondencias Automáticas Para L
- Correspondencia Para ES De L Concreto
- Correspondencias FLLUL ConESDeL valores heredados
- Correspondencias FLLUL
- Correspondencias De FL Concreta Para ES
- Correspondencias De FL ConcretaPara ES Concreta
- CorrespondenciasDeFLConcreta Para L
- Correspondencias Madre Para ES
- Correspondencias Para ES Con VALOR Concreto
- Correspondencias Para ES Concreta
- Correspondencias Para ES Concreta Con Sus L
- Correspondencias Para ES De L Concreto
- Correspondencias Para ES De vocablo Concreto
- Correspondencias Para L Con VALOR Concreto
- Correspondencias Para L Concreto
- Correspondencias Para L De ES Concreta
- Correspondencias Para vocablo Concreto
- FFLL Con Su Numero De Colocaciones
- FFLL Con Su Numero De L
- FFLL Con Su Numero De V
- FFLL Con X Numero De Colocaciones
- FFLL Con X Numero De L En Correspondencias
- FFLL Con X Numero De V En Correspondencias
- FFLL Sin Colocación
- FFLL Solapadas
- FFLL y L De Correspondencia
- FFLL y V De Correspondencias
- FL ES UL No Rechazados
- FL L UL Heredadas Rechazadas
- G De FL Heredadas De Es
- Heredados No Rechazados
- Identificadores De Lemas
- Informe Completo

- LEMAS Sin Colocaciones
- Nivel0 Raiz
- Nivel1
- Nivel2
- Nivel3
- Nivel4
- Nivel5
- Nivel6
- Nivel7
- Nivel8
- Nivel9
- Raiz
- Rasgos Heredados
- Rasgos Heredados De FL Concreta
- Valores Heredados Nuevos Para ES
- Valores Heredados Para ES
- Valores Heredados Para L
- Valores No Heredados
- Valores Nuevos Para Insertar
- Valores Para Insertar Iniciales
- Vocablos Repetidos En UL
- Get Todas FL
- Insertar rasgos heredados
- Insertar rasgos heredados de FL concreta
- Insertar rasgos heredados para L concreta
- Insertar valores heredados
- Insertar valores heredados de ES concreta
- Insertar valores heredados de ES (expr) concreta
- Insertar valores heredados nuevos para ES
- Insertar valores heredados nuevos para L
- Insertar valores heredados para ES
- Insertar valores heredados para L concreta
- Insertar valores heredados para L(expr) concreta
- Transformar glosas en UL



## ANEXO II – INSTALACIÓN DEL SISTEMA

La instalación que se va a describir a continuación se va a basar en un entorno Linux, en concreto para una instalación con la distribución Ubuntu 11. Sin embargo, es igualmente posible instalar la aplicación en un entorno Windows, por lo que a lo largo del documento se indicarán las diferencias entre la instalación en un entorno y en otro.

## 10.1. Instalación de los productos

Toda la instalación se basa en productos Open-Source con licencias que permiten su uso sin necesidad de pagar nada por ellos; a continuación se listan esos productos y las versiones utilizadas, así como direcciones de descarga de los mismos:

Apache-Tomcat 7.0	<a href="http://tomcat.apache.org/">http://tomcat.apache.org/</a>
MySQL 5.6	<a href="http://www.mysql.com/">http://www.mysql.com/</a>
SquirrelSQL 3.3.0	<a href="http://squirrel-sql.sourceforge.net/">http://squirrel-sql.sourceforge.net/</a>

En un entorno Ubuntu la manera más sencilla de realizar la instalación de estos productos es usando la herramienta apt (que viene instalada por defecto en el sistema). La instalación de Apache-Tomcat requiere la ejecución de los siguientes comandos:

```
sudo apt-cache search tomcat
sudo apt-get install tomcat7.0-webapps
sudo apt-get install tomcat7.0-admin
sudo apt-get install tomcat7.0
```

Estos comandos instalarán la versión 7.0 de Apache-Tomcat, además de los scripts necesarios en el directorio `/etc/init.d` para que se arranque automáticamente. Se debe comprobar que se ha configurado correctamente la variable de entorno `TOMCAT_HOME` con el comando

```
echo $TOMCAT_HOME
```

Que debería mostrar el valor `/opt/tomcat`. Si no aparece ningún valor, se debe establecer editando el archivo `~/.bash_profile` y añadiendo la línea:

```
export TOMCAT_HOME=/opt/tomcat
```

Y tras cerrar y abrir un nuevo terminal, se repite la comprobación.

La instalación de MySQL requiere primero instalar los paquetes de la misma:

```
sudo apt-get install mysql-server mysql-client
```

Para después establecer la contraseña de root de la base de datos (que por defecto es una cadena vacía):

```
sudo mysqladmin -u root -h localhost password 'mypassword'
sudo mysqladmin -u root -h myhostname password 'mypassword'
```

Sustituyendo `'mypassword'` por la contraseña requerida y `myhostname` por el nombre del servidor.

Finalmente se instala el cliente SQL para poder ejecutar más cómodamente los scripts, siempre y cuando se disponga de un escritorio gráfico en el sistema (en caso contrario lo podemos hacer en el



PC desde el que accedemos). La instalación simplemente consta de descomprimir el fichero descargado en el directorio donde se quiera instalar.

En entornos Windows, la instalación de estos productos se realiza a través de asistentes gráficos, por lo que simplemente se deben descargar las versiones apropiadas para ese sistema operativo y seguir las indicaciones de los asistentes.

## 10.2. Carga de base de datos.

La carga del contenido de la base de datos se realiza a través del cliente de base de datos que se ha descargado; para ello, se arranca el programa apareciendo la pantalla de la figura 10.1

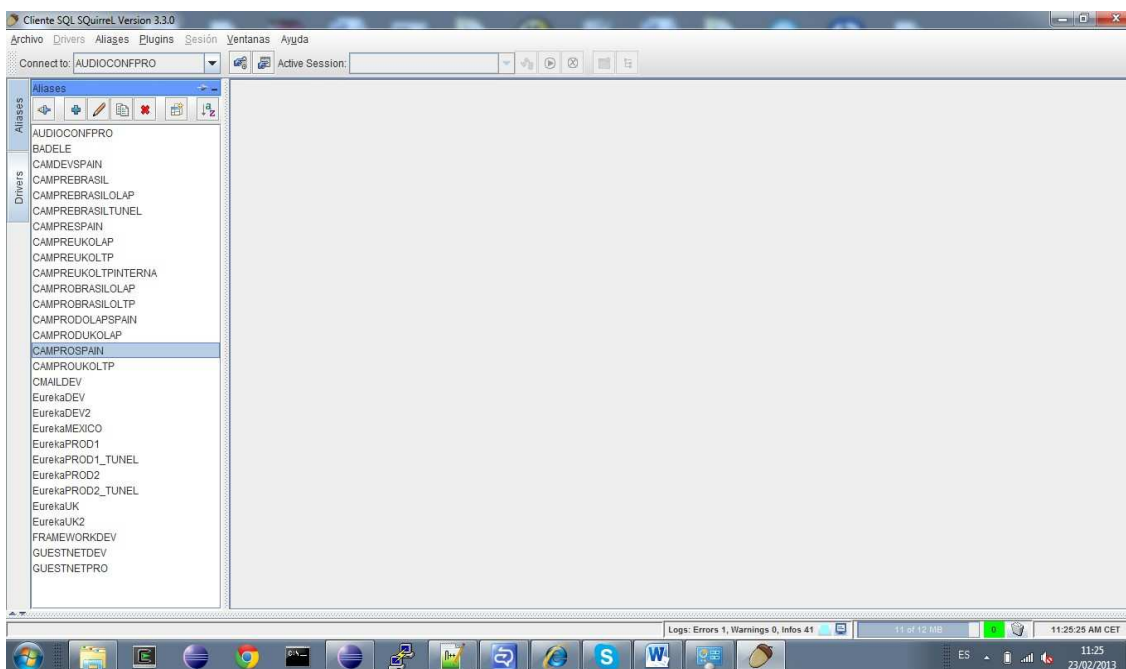


Figura 10.1 – Pantalla inicial del cliente de base de datos

Primero se debe registrar el driver JDBC de MySQL en la aplicación; para eso, se debe pinchar en 'Drivers', y seleccionar de la lista que aparece 'MySQL Driver'. Aparece un formulario en el que se debe marcar la pestaña 'Extra Class Path' y al pulsar 'Add' se debe seleccionar el fichero mysql-connector-java-5.1.15.-bin.jar de la carpeta 'lib' de la distribución de la aplicación.

Una vez registrado el driver, se debe crear la conexión con la base de datos. Para eso, primero hay que asegurarse de que el servidor mysql está ejecutándose; se abre un terminal y se ejecuta el comando:

```
service mysql status
```

Si la respuesta indica que el servidor no está iniciado, se debe arrancar ejecutando:

```
service mysql start
```

Hecho esto, se vuelve al cliente de base de datos y al pulsar el botón lateral 'Alias', se pulsa al

símbolo ‘+’ lo que abrirá un formulario como el de la figura 10.2

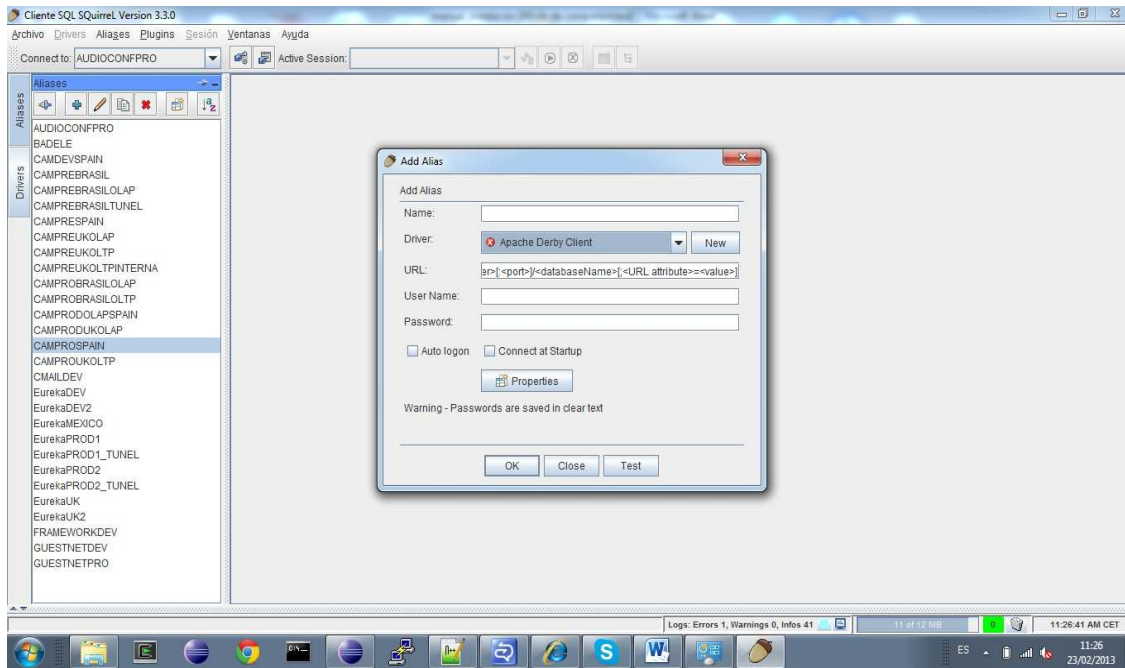


Figura 10.2 – Formulario de nuevo alias

En el que se deben introducir los siguientes datos:

NAME: BADELE

DRIVER: MySQL (registrado antes)

URL: jdbc:mysql://<<host>>/badele

USERNAME: root

PASSWORD: password (la introducida anteriormente al instalar el servidor de base de datos)

Con estos datos, se pulsa OK y después con el botón derecho encima de la conexión recién creada, se debe seleccionar ‘Connect’ lo que abrirá una ventana nueva con la conexión a la base de datos como la mostrada en la figura 10.3.

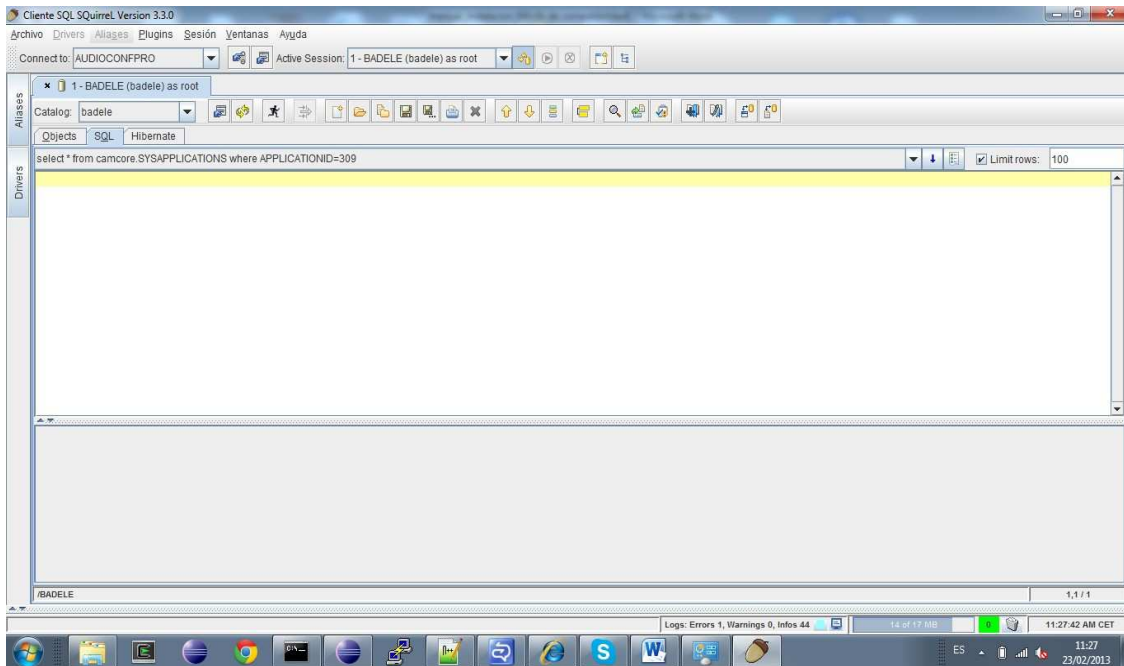


Figura 10.3. – Cliente de base de datos listo para usar

En esta pantalla, se copia el contenido del fichero dumpBadele5.sql, se marca entero pulsando Ctrl+A y se ejecuta pulsando AltGr+Intro. Esto ejecuta la carga de base de datos BADELE+ en el servidor MySQL. Tras esto, se puede cerrar el cliente de base de datos.

### 10.3. Despliegue de las aplicaciones

El despliegue de las aplicaciones se refiere a la instalación en el servidor tanto del servicio como de la aplicación web. En esta instalación se va a presuponer dos cosas:

- El servicio y la web estan en la misma máquina.
- Se van a utilizar los directorios indicados en este documento.

El despliegue de las aplicaciones se realiza exactamente igual en Windows, simplemente cambiando la notación de directorios de un sistema operativo por otro.

Para desplegar las aplicaciones, primero se deben crear directorios para la aplicación; para ello, se debe ejecutar:

```
mkdir /var/badele
mkdir /var/badele/data
mkdir /var/badele/tmp
mkdir /var/badele/config
```

Y copiar a este último directorio los ficheros

```
configService.properties
configWeb.properties
```

Comprobar si el servidor esta arrancado; para ello, se ejecuta el comando:

```
/etc/init.d tomcat status
```

Y si no esta iniciado,

```
/etc/init.d tomcat start
```

Una vez hecho esto, se copia el fichero service.zip al directorio \$TOMCAT\_HOME/webapps y se descomprime su contenido, lo que creará un directorio axis conteniendo el servicio. Para comprobar que el servicio esta iniciado, una vez realizado esto, se puede abrir un navegador en la propia máquina (o bien en el PC desde el que se haga la instalación) y acceder a la URL:

[http://<<ip\\_servidor>>:8080/axis/services/BADELEInterface?wsdl](http://<<ip_servidor>>:8080/axis/services/BADELEInterface?wsdl)

Si se ha realizado la instalación correctamente, debe aparecer en la pantalla del navegador el WSDL del servicio, como se muestra en la figura 10.4.

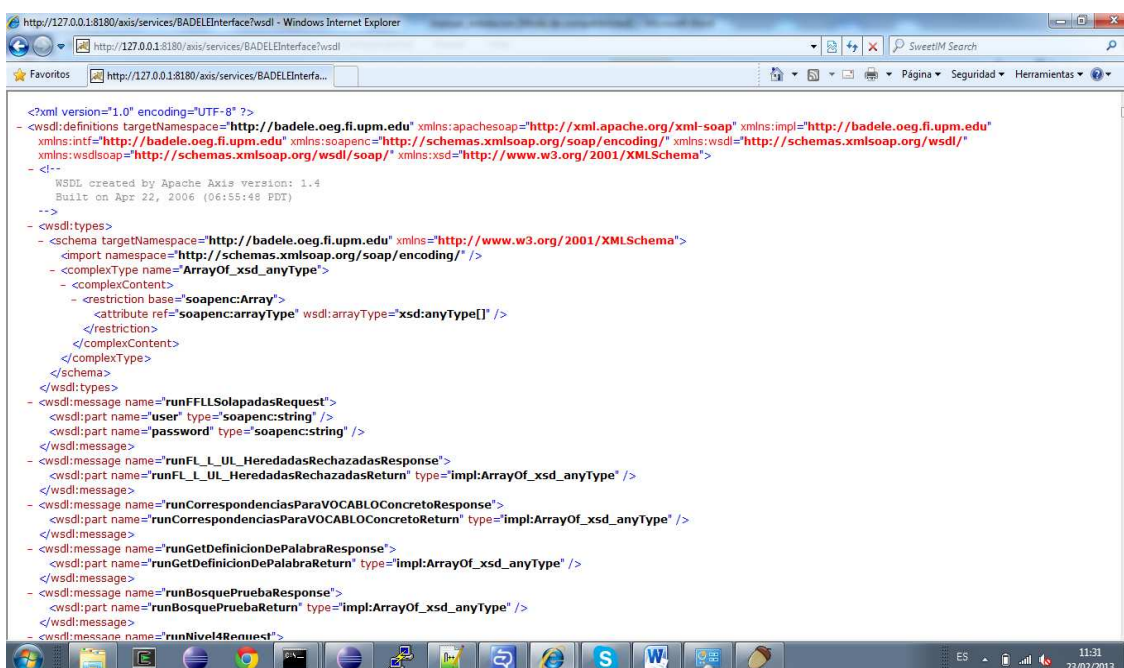


Figura 10.4. – WSDL del servicio instalado

La aplicación Web se instala copiando el fichero BADELEWeb.war al directorio \$TOMCAT\_HOME/webapps/, de manera que la aplicación se despliegue automáticamente y este ya disponible para usarse.

En el caso de que uno de los dos supuestos iniciales no se cumpla, puede ser necesario modificar la configuración de la aplicación web o del servicio.

### 10.3.1. Modificar la configuración de la aplicación web

En el fichero /opt/tomcat/webapps/BADELEWeb/WEB-INF/web.xml hay una sección como ésta:

```
<context-param>
    <param-name>config_file</param-name>
    <param-value>/var/badele/config/configWeb.properties</param-value>
</context-param>
```

En ella se indica dónde se encuentra el fichero de configuración, por lo que se puede cambiar la ubicación del mismo si es necesario. Por su parte, el propio fichero de configuración tiene el siguiente contenido:

```
serviceURL=http://127.0.0.1:8080/axis/services/BADELEInterface
folderXLS=/opt/tomcat/webapps/BADELEWeb/excels/
folderLanguages=/opt/tomcat/webapps/BADELEWeb/languages/
initializeServiceURL=http://127.0.0.1:8080/axis/Initializer?accessQ=pF1nrjnf34
```

El significado de cada uno de los parámetros es el siguiente:

- serviceURL: la URL del servicio, que sólo se debe cambiar si el servicio se instala en otra máquina.
- folderXLS: carpeta temporal para la generación de los ficheros excel.
- folderLanguages: carpeta donde se guardan ficheros con la traducción de textos de la web.
- initializeServiceURL: una URL especial del servicio que sirve para iniciarlo, y que (al igual que el primer parámetro) sólo se debería cambiar si el servicio y la web están en máquinas diferentes.

### 10.3.2. Modificar la configuración del servicio

El fichero /opt/tomcat/webapps/axis/WEB-INF/web.xml contiene la siguiente sección:

```
<context-param>
    <param-name>config_file</param-name>
    <param-value>/var/badele/config/configService.properties</param-value>
</context-param>
```

En ella se indica dónde se encuentra el fichero de configuración, por lo que se puede cambiar la ubicación del mismo si es necesario. Por su parte, el propio fichero de configuración tiene el siguiente contenido:

```
userbadele=userbadele
passwdbadele=passwdbadele
jdbc=jdbc:mysql://127.0.0.1:3306/badele?
user=root&password=pln&useOldAliasMetadataBehavior=true&autoReconnectForPools=true
maxnumrows=1000
```

El significado de los parámetros es el siguiente:

- userbadele: usuario de acceso al servicio por parte de una aplicación.
- passwdbadele: contraseña de acceso al servicio por parte de una aplicación.
- jdbc: cadena JDBC de acceso a la base de datos.
- maxnumrows: número máximo de filas que una consulta puede devolver.